

# HOME HEALTH CARE LOGISTICS PLANNING

A Thesis  
Presented to  
The Academic Faculty

by

Ashlea R. Bennett

In Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy in the  
H. Milton Stewart School of Industrial and Systems Engineering

Georgia Institute of Technology  
May 2010

# HOME HEALTH CARE LOGISTICS PLANNING

Approved by:

Associate Professor Alan L. Erera,  
Advisor  
H. Milton Stewart School of Industrial  
and Systems Engineering  
*Georgia Institute of Technology*

Professor Martin Savelsbergh  
H. Milton Stewart School of Industrial  
and Systems Engineering  
*Georgia Institute of Technology*

Professor Paul M. Griffin  
Harold and Inge Marcus Department of  
Industrial and Manufacturing  
Engineering  
*Pennsylvania State University*

Associate Professor Julie L. Swann  
H. Milton Stewart School of Industrial  
and Systems Engineering  
*Georgia Institute of Technology*

Professor of the Practice Mark  
Braunstein, M.D.  
College of Computing  
*Georgia Institute of Technology*

Date Approved: 8 December 2009

*“Whatever you do, work at it with all your heart, as working for the Lord, not for men,”*  
*Colossians 3:23.*

## ACKNOWLEDGEMENTS

There are many people whose support enabled the completion of this thesis. First, I would like to thank my advisor, Dr. Alan Erera, who taught me what it really means to conduct research. In addition to providing academic guidance, his frequent willingness to provide professional advice demonstrated his sincere interest in my success. I would also like to thank my committee members for their helpful suggestions: Dr. Martin Savelsbergh, Dr. Paul Griffin, Dr. Julie Swann, and Dr. Mark Braunstein.

I would like to thank the National Science Foundation and Visiting Nurse Hospice Atlanta for their financial support during my graduate studies. I am especially grateful to Mark Oshnock of Visiting Nurse, who provided opportunities for me to gain hands-on work experience in healthcare. I would like to thank David Cowan for connecting me with numerous healthcare professionals, but perhaps more importantly, for keeping me focused on things that truly matter.

I am tremendously thankful for my parents, who have always believed in me. At age five, they believed I was old enough for the piano and piano lessons I asked for, and most recently, never doubted I would complete this thesis. They have unceasingly supported my goals and given me their time and attention in quantities I can never repay.

Lastly, I would like to thank all of my friends and family who have been there for me throughout this process, especially: Amanda Lang and Lisa Delozier, for their prayers and wise counsel; Dylan Shepardson and Clarence Wardell, for the late night coffee shop work sessions; and my three year old nephew Jackson, for providing comic relief that only he was capable of during my most stressful days.

# TABLE OF CONTENTS

DEDICATION . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF TABLES . . . . .	viii
LIST OF FIGURES . . . . .	ix
SUMMARY . . . . .	xi
I INTRODUCTION . . . . .	1
II ROUTING PROBLEMS WITH FIXED APPOINTMENT TIMES . . . . .	7
2.1 Introduction . . . . .	7
2.2 FAP1( $\delta$ ): minimize duration of tour that serves all customers . . . . .	9
2.2.1 Impact of appointment times on tour duration for a given visit sequence . . . . .	10
2.2.2 TSP-based solution methods for FAP1( $\delta$ ) . . . . .	15
2.2.3 Network transformations for FAP1( $\delta$ ) . . . . .	19
2.3 FAP2( $\delta, T_{max}$ ): maximize number of customers served given tour duration constraint . . . . .	23
2.3.1 Problem statement . . . . .	23
2.3.2 Complexity of FAP2( $\delta, T_{max}$ ) . . . . .	25
2.3.3 Impact of appointment times on number of customers visited . . . . .	26
2.3.4 Customer subsets selected in optimal solutions to FAP2( $\delta, T_{max}$ ) . . . . .	30
2.3.5 TSP-based solution method for FAP2( $\delta, T_{max}$ ) . . . . .	31
2.3.6 Network transformation for FAP2( $\delta, T_{max}$ ) . . . . .	33
2.3.7 Concluding remarks regarding FAP2( $\delta, T_{max}$ ) . . . . .	35
2.4 Recommendations for static routing problems with fixed appointment times . . . . .	35
III HEURISTICS FOR DYNAMIC PERIODIC HOME HEALTH NURSE SCHEDULING PROBLEMS . . . . .	37
3.1 Introduction . . . . .	37
3.2 Literature review . . . . .	39
3.2.1 Home health nurse scheduling problems . . . . .	40
3.2.2 Periodic routing problems . . . . .	41

3.2.3	Dynamic routing problems . . . . .	43
3.3	Problem statement . . . . .	44
3.4	Rolling horizon myopic planning approach . . . . .	46
3.4.1	Route updating procedure . . . . .	49
3.4.2	Appointment time feasibility . . . . .	49
3.4.3	Admission appointment time feasibility and cost . . . . .	51
3.4.4	Regular appointment time feasibility and cost . . . . .	51
3.4.5	Appointment time combination feasibility and cost . . . . .	53
3.5	Distance heuristic . . . . .	54
3.6	Capacity heuristic . . . . .	57
3.6.1	Reduced visit capacity . . . . .	59
3.6.2	Reduced flexible capacity . . . . .	60
3.6.3	Daily workload capacity . . . . .	62
3.7	Experiments and results . . . . .	64
3.7.1	Absolute performance of the capacity heuristic . . . . .	66
3.7.2	Relative performance of the capacity and distance heuristics . . . . .	68
3.8	Conclusions and future work . . . . .	70
IV	HOME HEALTH NURSE DISTRICTING . . . . .	73
4.1	Introduction . . . . .	73
4.2	Literature review . . . . .	74
4.2.1	Location-allocation models for districting problems . . . . .	76
4.2.2	Set partitioning models for districting problems . . . . .	78
4.3	Problem statement . . . . .	81
4.4	Mathematical model . . . . .	82
4.4.1	District demand . . . . .	82
4.4.2	District cost . . . . .	82
4.4.3	District workload feasibility . . . . .	85
4.4.4	Set partitioning formulation for the nurse districting problem . . . . .	86
4.5	Solution approaches . . . . .	86
4.5.1	Initial solution . . . . .	87
4.5.2	Local search . . . . .	89

4.5.3	Column generation procedure . . . . .	93
4.6	Experiments and results . . . . .	102
4.6.1	Heuristic parameters . . . . .	104
4.6.2	Comparison of district cost approximations to simulated Hamiltonian path costs . . . . .	105
4.6.3	Results for 5-nurse instance, $TSP$ . . . . .	106
4.6.4	Results for 10-nurse instance, $TSP$ . . . . .	110
4.6.5	Results for 5-nurse instance, $SEP$ . . . . .	114
4.6.6	Absolute performance of the heuristics . . . . .	118
4.7	Conclusions . . . . .	118
V	FUTURE RESEARCH DIRECTIONS . . . . .	129
	BIBLIOGRAPHY . . . . .	131

## LIST OF TABLES

1	Arc costs for $\mathcal{G}$ . . . . .	30
2	Predecessor and successor times and locations of $q_8$ in Figure 7 . . . . .	50
3	Capacity heuristic output for $\alpha = 1.5$ , location distribution $U$ . . . . .	67
4	Visits per day, averaged across all arrival rates . . . . .	68
5	Visits per day and visit acceptance rates . . . . .	71
6	Average travel time per visit . . . . .	71
7	Recommendations for using the capacity and distance heuristics . . . . .	72
8	Select districting applications from the literature . . . . .	75
9	Parameters used in initial solution heuristic . . . . .	88
10	Characteristics of zip codes in test instance service region . . . . .	102
11	Best solution obtained using each heuristic, $TSP$ , $f = 5$ . . . . .	106
12	Moves of each type implemented and selected, $TSP$ , $f = 5$ . . . . .	107
13	Comparison of simulated hamiltonian path costs with $TSP$ approximation for heuristic solutions, $f = 5$ . . . . .	108
14	Comparison of simulated district costs and workloads with $TSP$ approximation for $CG+LS$ solution, $f = 5$ . . . . .	109
15	Column generation solution values and runtimes, $TSP$ , $f = 5$ . . . . .	110
16	Best solution obtained using each heuristic, $TSP$ , $f = 10$ . . . . .	112
17	Moves of each type implemented and selected, $TSP$ , $f = 10$ . . . . .	112
18	Column generation solution values and runtimes, $TSP$ , $f = 10$ . . . . .	114
19	Best solution obtained using each heuristic, $SEP$ , $f = 5$ . . . . .	116
20	Moves of each type implemented and selected, $SEP$ , $f = 5$ . . . . .	116
21	Comparison of simulated hamiltonian path costs with $SEP$ approximation for heuristic solutions, $f = 5$ . . . . .	116
22	Comparison of simulated district costs and workloads with $SEP$ approximation for $CG+BI$ solution, $f = 5$ . . . . .	117
23	Column generation solution values and runtimes, $SEP$ , $f = 5$ . . . . .	118



## LIST OF FIGURES

1	Example showing tightness of bound in Theorem 2.2.1 . . . . .	14
2	Counterexample for $NAT$ . . . . .	16
3	Tight example for $\delta = 2$ . . . . .	18
4	Asymmetric and undirected transformed network . . . . .	21
5	Example network, $T_{max} = 20$ , $\delta = 5$ . . . . .	31
6	Counterexample for $NAT2$ , $n_5 = 3$ , $c_1(T) = 12$ , $c_5(T) = 16$ . . . . .	33
7	Snapshot of master schedule at planning interval $g$ . . . . .	48
8	Master schedule illustrating need for distance tie breaker . . . . .	56
9	Selecting $q$ based on proximity to predecessor or successor patient location can reduce schedule capacity . . . . .	58
10	Example illustrating flexible schedule capacity . . . . .	61
11	Subunit boundary map and associated adjacency graph . . . . .	77
12	Daily nurse travel as Hamiltonian paths through random points in district .	83
13	Type I shift example . . . . .	90
14	Type II shift followed by type I shift example . . . . .	91
15	Swap example . . . . .	92
16	Append example and reject example . . . . .	97
17	Append/reject example . . . . .	99
18	Infeasible swap example . . . . .	100
19	Swap that could be created by append/reject . . . . .	101
20	District demand, workload, and cost vs. district area, $TSP$ , $f = 5$ . . . . .	107
21	Pricing problem execution time, $TSP$ , $f = 5$ . . . . .	111
22	District demand, workload, and cost vs. district area, $TSP$ , $f = 10$ . . . . .	113
23	Pricing problem execution time, $TSP$ , $f = 10$ . . . . .	115
24	$LS1$ solution map, $TSP$ , $f = 5$ . . . . .	120
25	$LS$ solution map, $TSP$ , $f = 5$ . . . . .	121
26	$CG + LS$ solution map, $TSP$ , $f = 5$ . . . . .	122
27	$BI$ solution map, $TSP$ , $f = 10$ . . . . .	123
28	$LS$ solution map, $TSP$ , $f = 10$ . . . . .	124

29	$CG + BI$ solution map, $TSP$ , $f = 10$ . . . . .	125
30	$BI$ solution map, $SEP$ , $f = 5$ . . . . .	126
31	$LS$ solution map, $SEP$ , $f = 5$ . . . . .	127
32	$CG + BI$ solution map, $SEP$ , $f = 5$ . . . . .	128

## SUMMARY

This thesis develops quantitative methods which incorporate transportation modeling for tactical and operational home health logistics planning problems. We define home health nurse routing and scheduling (HHNRS) problems, which are dynamic periodic routing and scheduling problems with fixed appointment times, where a set of patients must be visited by a home health nurse according to a prescribed weekly frequency for a prescribed number of consecutive weeks during a planning horizon, and each patient visit must be assigned an appointment time belonging to an allowable menu of equally-spaced times. Patient requests are revealed incrementally, and appointment time selections must be made without knowledge of future requests. First, a static problem variant is studied to understand the impact of fixed appointment times on routing and scheduling decisions, independent of other complicating factors in the HHNRS problem. The costs of offering fixed appointment times are quantified, and purely distance-based heuristics are shown to have potential limitations for appointment time problems unless proposed arc cost transformations are used. Building on this result, a new rolling horizon capacity-based heuristic is developed for HHNRS problems. The heuristic considers interactions between travel times, service times, and the fixed appointment time menu when inserting appointments for currently revealed patient requests into partial nurse schedules. The heuristic is shown to outperform a distance-based heuristic on metrics which emphasize meeting as much patient demand as possible.

The home health nurse districting (HHND) problem is a tactical planning problem which influences HHNRS problem solution quality. A set of geographic zones must be partitioned into districts to be served by home health nurses, such that workload is balanced across districts and nurse travel is minimized. A set partitioning model is formulated for HHND and a column generation heuristic is developed which integrates ideas from optimization and local search. Methods for estimating district travel and workload are developed and implemented within the heuristic, which outperforms local search on test instances.

## CHAPTER I

### INTRODUCTION

Home health care workers in the United States drive 5 billion miles each year to visit patients—double the number of miles traveled by United Parcel Service (UPS) drivers annually ([37], [2]). The logistics challenges associated with dispatching vehicles to deliver products to customer locations are very similar to those encountered when deploying nurses to deliver health care to patient homes, the latter being additionally complicated by medical constraints and patient service considerations. The research community has actively addressed these problems in the context of the freight transportation industry. However, the studies addressing their application in the home health industry are strikingly few ([3], [6], [9], [21], [40], [42]). In this thesis we make contributions to home health care logistics planning problems.

The home health industry is a critical component of the nation’s health care system that has the potential to lower the system-wide cost of delivery of care. Studies have shown that using home health to assist with the daily management of chronic disease decreases risk for hospitalizations [30]. Inside the hospital, care is estimated to cost \$1,889 per day [1]. Inside the home, a single visit costs \$132 [36]. Thus, while conservative estimates attribute only 3% of total health care expenditures to home care spending, the financial implications of shifting more care to the home setting are much more pronounced [38].

Realizing the benefits of this shift of delivery of care from hospital to the home will require a large home health nurse workforce. Effective utilization of these nurses will be key in meeting forecasted demand in the coming decades. The National Association for Home Care and Hospice estimates there were 210,000 licensed nurses and therapists employed in home care in 2007 [36]. The number of patients receiving home care services was 7.6 million. Demand for home care is expected to double by 2030 as the baby boomer generation ages, the number of chronic disease diagnoses increases, and the trend of shifting hospital care

to less acute settings gains momentum [43]. However, the supply of nursing services is not expected to keep pace with the increasing demand. By 2020, a 20% gap between the supply of skilled nurses and the demand for their services is expected [12]. One way to narrow this gap is to increase the productivity of the existing resources.

This thesis studies the problem of improving nurse productivity through the development of improved solution methodologies for both tactical and operational logistics planning problems encountered in the home health industry. At the tactical level, the assignment of home health nurses to geographic service regions is addressed. Resultant problems can be modeled as graph partitioning problems with various side constraints. At the operational level, the problem of developing daily visit schedules for home health nurses is studied. The optimization problems that arise in this context are dynamic variants of periodic routing and scheduling problems with fixed appointment times.

Fixed appointment times is a problem characteristic not addressed in the routing literature, but one which has an important application in home health nurse routing and scheduling problems. In routing problems with a fixed menu of appointment times, there is a set of  $\mathcal{N}$  customer locations, with each customer requiring a service time. A complete set of undirected arcs  $(i, j) \in \mathcal{A}$  connects all customer locations with known travel times that satisfy the triangle-inequality. Each customer must be assigned an appointment time belonging to an allowable menu of equally-spaced times  $\{a + k\delta : k = 0, \dots, m\}$ , where  $a$  is the earliest appointment time,  $m + 1$  is the total number of appointments, and  $\delta$  is the time elapsed between each allowable appointment.

Chapter 2 makes contributions to the routing literature in a study of static fixed appointment time routing problems. This chapter is motivated by the need to understand the impact of fixed appointment times on routing and scheduling decisions, independent of other complicating factors in the home health nurse scheduling problem. The costs of offering fixed appointment times are quantified under two separate objectives: (i) minimize the duration of a tour which serves all customers, and (ii) maximize the number of customers served by one capacity-constrained tour. We refer to resultant problems as FAP1( $\delta$ ) and FAP2( $\delta, T_{max}$ ), respectively, where  $T_{max}$  is a tour duration constraint. The effectiveness of

traditional distance-based routing heuristics for appointment time problems is studied, using both a traditional network and an alternative with transformed arc costs. The primary results of Chapter 2 are summarized below.

- Static fixed appointment time routing problems are defined and shown to be NP-hard, and special cases which are easy to solve are identified.
- Purely distance-based heuristics are shown to have potential limitations for appointment time problems. Developing tours which minimize the travel distance required to serve a fixed set of customers without considering the interaction of the appointment time grid with the travel and service times can result in tours having twice the duration of the minimum duration fixed appointment time tour. This observation provides some of the motivation for a new solution method for a dynamic and periodic fixed appointment time problem in Chapter 3.
- A simple arc cost transformation is shown to enable the use of standard heuristics for fixed appointment time routing problems in certain cases. Additionally, we show that orienteering problem (OP) solution methods can be used to solve  $FAP2(\delta, T_{max})$ .

Chapter 3 studies a fixed appointment time routing problem that is additionally complicated by dynamic and periodic components. Appointment time selections for each customer must repeat with some pre-specified frequency, and must be made without knowledge of future arriving customers. This problem is faced by home health agencies that must make scheduling decisions for new patients requests as they arrive. The heuristics and network transformations presented in Chapter 2 do not sufficiently address the additional complications inherent in this problem variant. However, the ideas can be extended to rolling horizon planning heuristics which are appropriate for dynamic periodic routing problems. A rolling horizon myopic planning approach is developed for a single-vehicle routing problem with a dynamic customer set. Building on the result that a purely distance-based heuristic is not likely to perform well for fixed appointment time routing problems, a new capacity-based heuristic (CH) for inserting customer requests is developed and implemented within the

rolling horizon framework. At each planning period, the heuristic explicitly considers remaining available time in the vehicle’s schedule when inserting currently revealed requests in an attempt to preserve capacity for inserting future customer requests.

The dynamic periodic fixed appointment time problem studied in Chapter 3 is motivated by its application to home health nurse routing and scheduling. Thus, the chapter defines home health nurse routing and scheduling (HHNRS) problems for a set of patients that need to be visited according to a prescribed weekly frequency for a prescribed number of consecutive weeks during a planning horizon. Each visit to the patient must be assigned a precise appointment time, chosen from a fixed menu of equally-spaced appointment times, such as  $\{8:00, 8:15, 8:30, \dots\}$ . This problem formulation constitutes a contribution to the home health literature, which does not address the complicating dynamic and periodic aspects of the home health nurse scheduling problem.

The rolling horizon myopic planning approach developed in Chapter 3 is implemented in the context of the HHNRS problem. A computational study is performed to compare the capacity heuristic against a traditional distance-based heuristic (DH) on various problem instances modeled after real home health applications. The primary results of the study are below.

- DH outperforms CH on metrics which emphasize minimizing distance traveled. On average, the schedules produced by DH require 8% less travel per patient visit than those produced by CH.
- DH is not able to use savings in travel time to visit additional patients under fixed appointment time constraints. CH outperforms DH on the primary metrics, which emphasize meeting as much patient demand as possible. On average, the schedules produced by CH accept 4% more patients and perform 4% more visits per day than schedules produced by DH. Across a home health workforce of 210,000 nurses and therapists, this constitutes an additional 12.6 million patient visits per year.

The focus throughout Chapter 3 is on the single-nurse problem because in practice, the service regions of home health agencies are partitioned into smaller districts to be served by

a single nurse or small subset of nurses. In Chapter 4, we study how the geographic areas (*e.g.*, zip codes) served by a single home care agency should be grouped into a set of home health nurse service districts. Two primary considerations include balancing workload across districts and minimizing nurse travel between patient visits. We model the home health nurse districting (HHND) problem as a graph partitioning problem with side constraints. The nodes in the graph represent zip code service areas and have a weight associated with some workload measure; for example, patient count or patient visit count. The arcs represent adjacencies between zip codes and may have an associated cost; for example, centroid-to-centroid distances. A solution to the home health nurse districting problem partitions zip codes into contiguous districts, such that the workload of each district is within the allowable bounds of the nurse or nurses serving the district, and some measure of district compactness is maximized. Chapter 4 develops an optimization-based heuristic for HHND. The primary contributions of the chapter are as follows.

- Two methods for measuring district compactness and workload are developed, which rely on approximations of the expected travel to serve patient demand originating within each region.
- A model is developed which can be used to provide managerial insight to various home care strategic planning decisions. For example, should districts be designed to be served by a single nurse, or by a team of  $f > 1$  nurses? What are the implications of balancing patient visit count across regions instead of balancing expected workload? In what geographic areas would it be beneficial to grow the business?
- The additional value provided by optimization-based heuristics over local search methods is quantified for home health nurse districting problems. On test instances studied, optimization-based heuristics improve initial feasible solutions by an average of 5%, and improve the best solutions obtained using local search methods by an average of 3.5%.

A solution to HHND provides a set of nurse service districts that allows the overall scheduling problem for a home care organization to be decomposed into separate HHNRS



subproblems for each district. For the case where districts are served by a single nurse, the methodology presented in Chapter 3 can be used to develop daily nurse schedules for each district developed using the methodology in Chapter 4. We plan to integrate the two solution mechanisms in the future, and extend the HHNRS methodology to the multiple-nurse problem. Chapter 5 discusses avenues for future research.

## CHAPTER II

### ROUTING PROBLEMS WITH FIXED APPOINTMENT TIMES

#### 2.1 *Introduction*

The study of fixed appointment time routing problems is motivated by businesses which must visit their customers at fixed times when the customer is required to be present. This is common in service industries such as home health nursing, cable television repair, and home grocery delivery. In some such industries, common practice is to give customers a time window during which they can expect the service visit (or pickup/delivery). Unfortunately, these windows are often as long as a half day to a full day. Customer service can be improved by shortening the length of the time window by giving the customer, for example, a 1-hour time window instead of a 4-hour window. An even greater customer service improvement would result if the customer were given an exact time to expect the visit. Using a fixed menu of equally-spaced appointment times (e.g., 1:00 pm, 1:15 pm, 1:30 pm, etc.) ensures that customers will not be assigned “irregular” times such as 1:37 pm. A company that implements equally-spaced appointment times has the opportunity to improve customer service, but they do so at a cost of reduced flexibility in their routes. Because the appointment times at customers are constrained to start at fixed times, idle time will be induced if the service person arrives to a customer location before the corresponding appointment time. The time spent waiting decreases the total time the service-person has available to serve customers. A company considering such a policy will need an accurate estimate of the costs and benefits associated with offering fixed appointment times. In this chapter, we evaluate these costs and develop heuristics for scheduling customers at fixed appointment times.

In static routing problems with fixed appointment times, there is a set of  $\mathcal{N}$  customer locations, with each customer  $i \in \mathcal{N}$  requiring service time  $w_i \geq 0$ , and all requests are

assumed to be served by a single mobile resource that we will denote the vehicle. All customer requests are known in advance. A complete set of undirected arcs  $(i, j) \in \mathcal{A}$  connects all customer locations. The travel time  $h_{ij} \geq 0$  on each arc is known and the network  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$  satisfies the triangle-inequality. Each customer is assigned an appointment time from a menu of equally-spaced times  $\{a + k\delta : k = 0, \dots, m\}$ , where  $a$  is the earliest appointment time,  $m + 1$  is the total number of appointments, and  $\delta$  is the time elapsed between each allowable appointment time (also referred to as the grid spacing parameter). The problem is to assign a set of feasible appointment times to customer requests. A set of appointment times is **feasible** if the vehicle can serve the requests in the order specified without arriving late to any of the customer visits. In some cases, feasibility may also require the total duration of the tour to be below some limit. Here, tour duration requires further explanation.

Let 0 denote a depot where the vehicle is based. Let  $F$  be the customer with the earliest appointment time  $q_F$ , such that it will be the first customer visited. We assume that the vehicle is able to leave the depot in order to arrive on time at customer  $F$ . Let  $L$  be the customer with the latest appointment time  $q_L$ , such that it will be the last customer visited. A **depot-based appointment time tour (T)** is defined as follows.

**Definition 1.** *A depot-based appointment time tour begins when the vehicle leaves the depot at time  $(q_F - h_{0F})$  to arrive at the first customer  $F$  at appointment time  $q_F$ . The vehicle then visits customers in sequence, with each visit occurring at an appointment time. The tour ends when the vehicle returns to the depot at time  $q_L + w_L + h_{L0}$  after serving the last customer at appointment time  $q_L$ .*

Note that while service at each customer must begin at an appointment time, the vehicle is not required to leave and return to the depot at an appointment time. There is no fixed shift start and end time because the depot typically represents the home location of the resource (e.g., nurse, repair technician) in the primary application areas for which this study is designed. The duration,  $c(T)$ , of a depot-based appointment time tour is calculated using Equation (1):

$$c(T) = q_L + w_L + h_{L0} - q_F - h_{F0}. \quad (1)$$

Using this definition of tour duration, we study the impact of fixed appointment times for static routing problems with two different objectives. Specifically, we study the following two Fixed Appointment Time (FAP) problems on a network  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ .

**FAP1( $\delta$ ):** Minimize the duration of a depot-based appointment time tour with grid spacing parameter  $\delta$ , such that the tour serves all customers.

**FAP2( $\delta, T_{max}$ ):** Given a tour duration constraint  $T_{max}$ , maximize the number of customers that can be served via a minimum duration depot-based appointment time tour with grid spacing parameter  $\delta$ .

Sections 2.2 and 2.3 present findings for FAP1( $\delta$ ) and FAP2( $\delta, T_{max}$ ). Unless otherwise stated, we use the following set of **assumptions** in this chapter.

- All problem data are non-negative integers ( $w_i, \delta, q_i, T_{max}, h_{ij}$ ). Travel times  $h_{ij}$  are strictly positive.
- Service at each customer must begin at an appointment time, but the vehicle is not restricted to leave and return to the depot at an appointment time.
- Each tour is such that the earliest feasible appointment time has been assigned to each customer, given the sequence of customers served.
- When tour duration is not limited by an upper bound  $T_{max}$ , there are a sufficient number of appointments available to allow serving all customer requests.

## 2.2 *FAP1( $\delta$ ): minimize duration of tour that serves all customers*

Minimizing the time required to visit a set of customers is a common objective in many freight routing applications, and in fact, in any application which requires visiting a set of locations in sequence. Here, we study the objective of minimizing tour duration under the constraint that each customer must be visited at precise appointment times, where each

appointment time is selected from a menu of equally-spaced times. We refer to this problem, where the singular objective is to minimize tour duration, as  $\text{FAP1}(\delta)$ .

Fixed appointment times may represent a potential customer service improvement. As such, any organization wishing to offer this service will need an understanding of the costs associated with doing so. In this section, we first compare the duration of a given tour when fixed appointment times are used versus when they are not. Clearly, a tour that meets fixed appointment time constraints can have duration no shorter than a tour which does not enforce those constraints. We develop an upper bound on the ratio by which the duration of the appointment time tour may exceed the unrestricted tour.

One “naive” approach to find a minimum duration appointment time tour is to first solve a TSP on a given network, then transform the resulting tour to an appointment time tour with grid spacing parameter  $\delta$ . We show that this heuristic is not guaranteed to solve  $\text{FAP1}(\delta)$ , and we provide a worst case bound for its performance. Finally, we develop a simple network transformation that enables the use of TSP-based solution methods for obtaining optimal solutions to  $\text{FAP1}(\delta)$ .

### 2.2.1 Impact of appointment times on tour duration for a given visit sequence

Suppose we have identified a tour serving  $n$  customers, where a tour is defined to be an ordering of the customer visits. Also, suppose that the tour is not constrained in total duration. We compare the duration of the tour to the duration of a depot-based appointment time tour visiting the customers in the same sequence. In this section, we show that the duration of the depot-based appointment time tour can be  $\delta$  times higher than the unrestricted tour.

Note that a set of feasible appointment time assignments yielding a minimum duration appointment time tour can be derived from a specified tour, and conversely, a tour can be derived from a set of feasible appointment time assignments.

- **Derive appointment time assignments from specified tour:** Assign the earliest appointment time from the menu of available appointment times to the first customer following the depot in the sequence specified by the tour. To each remaining customer

in sequence, assign the earliest feasible appointment time. The earliest feasible appointment time for customer  $j$  which is immediately preceded by a visit to customer  $i$  is computed using Equation (2):

$$q_{i+1} = \left\lceil \frac{q_i + w_i + h_{i,i+1}}{\delta} \right\rceil \delta = q_i + \left\lceil \frac{w_i + h_{i,i+1}}{\delta} \right\rceil \delta. \quad (2)$$

- **Derive tour from specified appointment time assignments:** Leave the depot at time  $q_F - h_{0F}$  and travel to the first customer. Visit the remaining customers in order of increasing appointment time. Return to the depot after all customers have been visited.

The total duration of a tour  $T$  visiting  $n$  customers under a fixed appointment time grid with spacing  $\delta$  can be calculated using Equation (3):

$$c_\delta(T) = h_{01} + \sum_{i=1}^{n-1} \left\lceil \frac{w_i + h_{i,i+1}}{\delta} \right\rceil \delta + w_n + h_{n0}. \quad (3)$$

Note that in addition to travel time and service time, the tour duration could also include idle time. For patient  $i+1$  visited after patient  $i$ , if the service time  $w_i$  plus the travel time  $h_{i,i+1}$  is not a multiple of  $\delta$ , idle time will be induced before service at  $i+1$  can begin. Let the duration of the idle time induced between  $i$  and  $i+1$  in an appointment-time tour be denoted as  $I_{i,i+1}$ , calculated using Equation (4):

$$I_{i,i+1} = q_j - (q_i + w_i + h_{i,i+1}) = \left\lceil \frac{w_i + h_{i,i+1}}{\delta} \right\rceil \delta - w_i - h_{i,i+1}. \quad (4)$$

Consider an appointment-time grid with parameter  $\delta = 1$ . Under the assumption of data integrality, the duration of the depot-based appointment time tour reduces to  $c_1(T)$ :

$$c_1(T) = h_{01} + \sum_{i=1}^{n-1} (w_i + h_{i,i+1}) + w_n + h_{n0}. \quad (5)$$

This duration only includes travel time and service time, and is equivalent to the duration of a tour which does not use appointment times. Throughout the remainder of this chapter,  $\delta = 1$  is used to represent the unrestricted case.

In this section we establish a bound on the ratio of  $c_\delta(T)$  to  $c_1(T)$ . Only the idle time potentially included in  $c_\delta(T)$  differentiates it from  $c_1(T)$ . To see this, note that Equation (3) can be rewritten as Equation (6), and then  $c_\delta(T)$  can be expressed in terms of  $c_1(T)$ :

$$c_\delta(T) = h_{01} + \sum_{i=1}^{n-1} (w_i + h_{i,i+1} + I_{i,i+1}) + w_n + h_{n0}, \quad (6)$$

$$c_\delta(T) = c_1(T) + \sum_{i=1}^{n-1} I_{i,i+1}. \quad (7)$$

Establishing a bound on the ratio of the durations  $c_\delta(T)$  and  $c_1(T)$  requires establishing a bound on the amount of idle time included in  $c_\delta(T)$ . Lemma 2.2.1 establishes a bound on idle time.

**Lemma 2.2.1.** *Let the  $n$  customer locations visited by a depot-based appointment time tour  $T$  with parameter  $\delta > 1$  be indexed by the order in which they are visited. Under the assumptions of data integrality and strictly positive travel times, the total amount of idle time induced between all pairs of consecutive patient visits in  $T$  is bounded by:*

$$0 \leq \sum_{i=1}^{n-1} I_{i,i+1} \leq (n-1)(\delta-1). \quad (8)$$

*Proof.* In a minimum duration depot-based appointment time tour with parameter  $\delta > 1$ , it is clear that the idle time between any two customer appointments must be bounded above by the spacing between consecutive available appointment times,  $\delta$ . Because all customer requests are known *a priori*, there is no advantage to waiting longer than necessary to begin service at some customer location. Thus, the earliest feasible appointment time is selected for each customer. Furthermore, when the assumptions of data integrality and strictly positive travel times hold, the idle time between any two consecutive customer visits is bounded above by  $\delta - 1$ .

In a depot-based appointment time tour serving  $n$  customers, idle time can be induced between at most  $n - 1$  pairs of consecutive customer locations:  $(1, 2), (2, 3), \dots, (n - 1, n)$ . Idle time is not induced between the depot and the first and last customer locations because

the vehicle is not required to leave and return to the depot at an appointment time. Thus, we have established the upper bound on the total idle time in  $T$ .

To establish the lower bound of zero, note it is possible for a depot-based appointment time tour to include no idle time. For example, when  $w_i + h_{i,i+1}$  is a multiple of  $\delta$  for all  $i = 1, \dots, n - 1$ , no idle time is induced.

□

By combining the bound established on idle time in Lemma 2.2.1 with Equation (7) which states  $c_\delta(T)$  in terms of  $c_1(T)$ , we can establish Theorem 2.2.1.

**Theorem 2.2.1.** *For a given tour  $T$  and integer  $\delta \geq 1$ ,  $c_\delta(T)$  must be greater than or equal to  $c_1(T)$ , and the ratio of the two durations is bounded by:*

$$1 \leq \frac{c_\delta(T)}{c_1(T)} \leq 1 + \frac{(n-1)(\delta-1)}{n+1}. \quad (9)$$

*Proof.* To see that the ratio is bounded below by 1, note that  $c_\delta(T) = c_1(T)$  when  $c_\delta(T)$  includes no idle time. Equation (10) establishes an upper bound on the numerator of Equation (9), and follows directly from Equation (7) and Lemma 2.2.1:

$$c_\delta(T) \leq c_1(T) + (n-1)(\delta-1). \quad (10)$$

Given integral problem data with strictly positive travel times, the duration of any tour with parameter  $\delta \geq 1$  is at least  $n + 1$ , the number of travel segments required to visit  $n$  customers and the depot. Thus, we have a lower bound of  $n + 1$  on the denominator of Equation (9). Equation (11) combines the results:

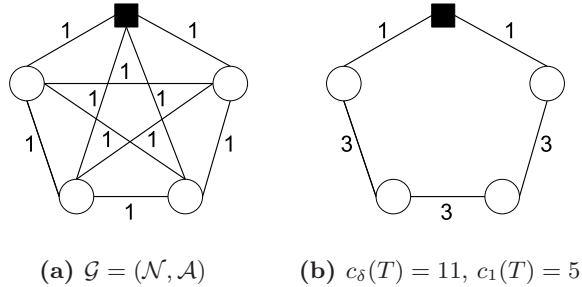
$$1 \leq \frac{c_\delta(T)}{c_1(T)} \leq \frac{c_1(T) + (n-1)(\delta-1)}{c_1(T)} \leq 1 + \frac{(n-1)(\delta-1)}{n+1}. \quad (11)$$

□

We can easily construct an example where this bound is tight. Consider Figure 1a where the square represents the depot. Let  $\delta = 3$ ,  $w_i = w = 0$ , and  $h_{ij} = h = 1$ . Consider the tour shown in Figure 1b. Its duration with no appointment times is 5. With an appointment



time grid with parameter  $\delta = 3$ , the three travel segments which do not connect to the depot will incur 2 units of idle time each. The resulting duration of the depot-based appointment time tour is 11, and the ratio of the durations is  $\frac{11}{5}$ .



**Figure 1:** Example showing tightness of bound in Theorem 2.2.1

Corollary 2.2.2 is established by taking the limit of the result in Theorem 2.2.1 as the number of customers grows large.

**Corollary 2.2.2.** *As  $n \rightarrow \infty$ ,  $\frac{c_\delta(T)}{c_1(T)}$  converges exactly to  $\delta$ .*

*Proof.*

$$\lim_{n \rightarrow \infty} \frac{c_\delta(T)}{c_1(T)} = \lim_{n \rightarrow \infty} 1 + \frac{(n-1)(\delta-1)}{n+1} = \delta. \quad (12)$$

□

In this section, we developed a bound on the ratio of tour durations when appointment times are used versus when they are not for a given sequence of  $n$  customer visits on network  $\mathcal{G}$ . Recall that under integral problem data, a grid spacing parameter of  $\delta = 1$  results in a tour which includes travel time and visit time but no idle time and is equivalent to the unrestricted case. Observe that finding the tour which minimizes  $c_1(T)$  is equivalent to solving the TSP on  $\mathcal{G}$ . The visit sequence which minimizes  $c_1(T)$  on a set of  $n$  customers also minimizes total travel time, because the total visit time across  $n$  customers is independent from the order in which customers are visited.

Suppose  $T_1$  is the optimal TSP on  $\mathcal{G}$ . Then,  $T_1$  is also an optimal solution to FAP1(1) on  $\mathcal{G}$ . The bound given in Equation (13) on the duration of  $T_1$  with a grid spacing parameter of  $\delta > 1$  follows directly from Theorem 2.2.1:

$$c_\delta(T_1) \leq c_1(T_1) \left( \frac{(n-1)(\delta-1)}{n+1} \right). \quad (13)$$

The duration of the TSP visit sequence under  $\delta > 1$  can be almost  $\delta$  times higher than its unrestricted duration. Let  $T_\delta$  represent the optimal solution to  $\text{FAP1}(\delta)$  for some  $\delta > 1$ . Theorem 2.2.1 does not describe how  $c_\delta(T_1)$  compares to  $c_\delta(T_\delta)$ , the *optimal* solution to  $\text{FAP1}(\delta)$ . Thus, in the next section, we explore the effectiveness of a TSP-based heuristic for  $\text{FAP1}(\delta)$ . We also develop a worst-case performance ratio for the heuristic that provides information regarding the ratio  $\frac{c_\delta(T_\delta)}{c_1(T_1)}$ .

### 2.2.2 TSP-based solution methods for $\text{FAP1}(\delta)$

In this section, we present a naive TSP-based heuristic for  $\text{FAP1}(\delta)$ . We show that while it is obvious that this heuristic is guaranteed to find optimal solutions to  $\text{FAP1}(1)$ , it is not guaranteed to find optimal solutions to  $\text{FAP1}(\delta)$  for general  $\delta$ .

Heuristic *NAT* is given in Algorithm 1. This heuristic first solves a TSP on  $\mathcal{G}$  without considering appointment times and then sets the earliest feasible appointment times for each customer in the sequence specified.

---

**Algorithm 1** *NAT*: TSP-based heuristic for  $\text{FAP1}(\delta)$

---

- 1: Obtain  $T_1 = \{0-i_1-i_2-\dots-i_n-0\}$  by solving a TSP on  $\mathcal{G}$ .
  - 2: Set  $q_{i_1} = a$
  - 3: **for all**  $l = 2, \dots, n$  **do**
  - 4:   Set  $q_{i_l} = q_{i_{l-1}} + \left\lceil \frac{w_{i_{l-1}} + h_{i_{l-1}, i_l}}{\delta} \right\rceil \delta$
  - 5: **end for**
- 

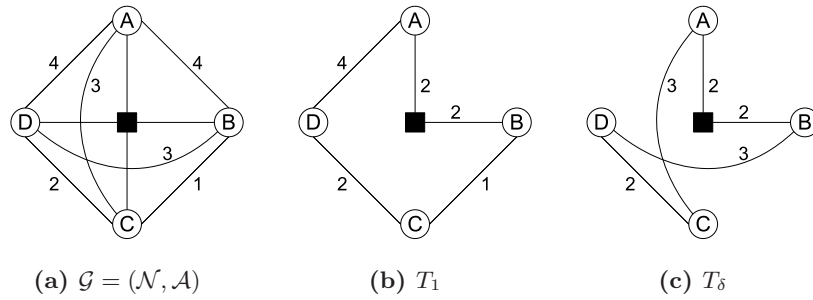
Clearly, *NAT* produces feasible solutions to  $\text{FAP1}(\delta)$ . However, the solutions are not necessarily optimal. A counterexample to illustrate this is provided in Section 2.2.2.1. Then, a worst-case performance bound for this approach is developed in Section 2.2.2.2.

#### 2.2.2.1 Counterexample for *NAT*

Although *NAT* provides optimal solutions to  $\text{FAP1}(1)$ , we can easily construct an example problem instance in which *NAT* does not produce the optimal solution to  $\text{FAP1}(\delta > 1)$ . Consider the example shown in Figure 2a. The depot is represented by the rectangle. The

length of the arcs connecting the depot to all customer locations is 2, and the length of the arcs connecting customer locations to each other are shown. Let  $w_i = w = 0$  for all customers, and let  $\delta = 3$ .

An optimal solution to the TSP and FAP1(1) is shown in Figure 2b. It has TSP duration of 11, and a duration of 16 once it has been transformed to a depot-based appointment time tour with parameter  $\delta = 3$ . An optimal solution to FAP1(3) is shown in Figure 2c. It has appointment-time duration of 13. Thus, the tour obtained via *NAT* is not the optimal solution to FAP1(3).



**Figure 2:** Counterexample for *NAT*

#### 2.2.2.2 Worst-case performance ratio for *NAT*

In this section, we establish a worst-case performance ratio for *NAT* over all possible instances of FAP1( $\delta$ ). Let  $Z_\delta^{NAT}(I)$  be the total cost of the solution produced by *NAT* on instance  $I$ , and let  $Z_\delta^*(I)$  be the total cost of the optimal solution to FAP1( $\delta$ ) on instance  $I$ .

**Theorem 2.2.3.** *For all instances  $I$  of FAP1( $\delta$ ) with appointment time grid spacing parameter  $\delta$ ,  $\frac{Z_\delta^{NAT}(I)}{Z_\delta^*(I)} < 2 - \frac{1}{\delta}$ .*

*Proof.* Continuing with earlier notation, let  $T_1$  be the optimal TSP tour and  $T_\delta$  be the optimal solution to FAP1( $\delta$ ). To prove Theorem 2.2.3, we must show that Equation (14) holds:

$$\max_I \frac{Z_\delta^{NAT}(I)}{Z_\delta^*(I)} = \max_I \frac{c_\delta(T_1)}{c_\delta(T_\delta)} \leq 2 - \frac{1}{\delta}. \quad (14)$$

From Equation (10), we have that  $c_\delta(T_1) \leq c_1(T_1) + (n-1)(\delta-1)$ . By optimality of  $T_1$  for the TSP objective, we know  $c_1(T_1) \leq c_1(T_\delta)$ . We also have that  $c_1(T_\delta) \leq c_\delta(T_\delta)$ , by applying the lower bound of 1 given on the ratio of  $\frac{c_\delta(T)}{c_1(T)}$  in Theorem 2.2.1. Thus, we have an upper bound of  $c_\delta(T_\delta) + (n-1)(\delta-1)$  established for the numerator of Equation (14).

To establish a lower bound on the denominator of Equation (14), note that  $c_\delta(T_\delta)$  must be greater than  $(n-1)\delta$ . In a tour which visits  $n$  customers and the depot, each of the  $n$  customer visits must be separated by at least  $\delta$  units of time. Combining the results gives the following:

$$\begin{aligned} \max_I \frac{c_\delta(T_1)}{c_\delta(T_\delta)} &\leq \frac{c_1(T_1) + (n-1)(\delta-1)}{c_\delta(T_\delta)} \leq \frac{c_1(T_\delta) + (n-1)(\delta-1)}{c_\delta(T_\delta)} \\ &\leq \frac{c_\delta(T_\delta) + (n-1)(\delta-1)}{c_\delta(T_\delta)} < 1 + \frac{(n-1)(\delta-1)}{(n-1)\delta} = 2 - \frac{1}{\delta}. \end{aligned} \quad (15)$$

□

In the next section, we provide an example showing the bound in Theorem 2.2.3 is tight for certain values of  $\delta$ .

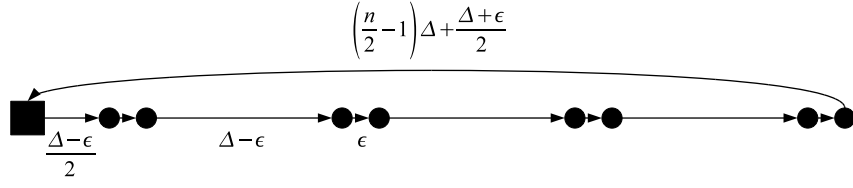
### 2.2.2.3 Tight example for NAT worst case performance ratio

Figure 3 is an example showing the bound in Theorem 2.2.3 is tight. In the example network, there are  $m$  pairs of customer locations and a depot. Each customer is separated from its sibling by  $\epsilon$ , and the spacing between pairs of customers is  $\delta - \epsilon$ . The distance from the depot to the first customer location is  $\frac{(\delta-\epsilon)}{2}$ . An optimal TSP tour ( $T_1$ ) and an optimal appointment time tour ( $T_\delta$ ) and their durations are shown in Figures 3a and 3b. Taking the limit of the ratio between  $c_\delta$  and  $c_1$  as the number of customers grows large, we obtain:

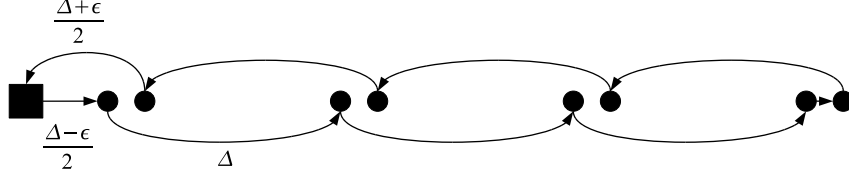
$$\lim_{m \rightarrow \infty} \frac{(3m-1)\delta}{2m\delta} = \frac{3}{2}. \quad (16)$$

Thus, for the case of  $\delta = 2$ , this example shows the bound expression in Theorem 2.2.3 is tight.

We have seen that solving a TSP on  $\mathcal{G}$  does not in general solve FAP1( $\delta$ ). Solving a TSP and then transforming the TSP tour to a depot-based appointment time tour produces



(a)  $T_1$ :  $c_1(T_1) = (2m - 1)\delta + \epsilon$ ,  $c_\delta(T_1) = (3m - 1)\delta$



(b)  $T_\delta$ :  $c_1(T_\delta) = (2m - 1)\delta + \epsilon$ ,  $c_\delta(T_\delta) = 2m\delta$

**Figure 3:** Tight example for  $\delta = 2$

tours with durations up to  $2 - \frac{1}{\delta}$  times higher than optimal appointment time tours.

#### 2.2.2.4 Note on tour obtained in Step 1 of heuristic NAT

Note that heuristic *NAT* assumes we have access to the optimal TSP tour. Suppose that we do not have an approach that yields the optimal tour; this is clearly more likely as  $n$  becomes large. Consider a new heuristic *NAT* – *C*, which only differs from *NAT* in that it uses Christofides' heuristic to obtain a tour in Step 1. Let  $T_1$  represent the optimal TSP tour,  $T_{1C}$  represent the tour obtained via Christofides' heuristic, and  $T_\delta$  represent the optimal depot-based appointment time tour. To develop a worst-case performance bound for *NAT* – *C*, the logic from the proof of Theorem 2.2.3 can be reapplied. Additionally, we need the following result from Christofides [16].

**Theorem 2.2.4.** *For all instances of the TSP satisfying the triangle inequality,  $c_1(T_{1C}) \leq \frac{3}{2}c_1(T_1)$ .*

Using Theorem 2.2.4, we can obtain the following result:

$$c_1(T_{1C}) \leq \frac{3}{2}c_1(T_1) \leq \frac{3}{2}c_1(T_\delta). \quad (17)$$

Then, the worst-case performance bound for  $NAT$  can be modified for  $NAT - C$ , as shown in Theorem 2.2.5. However, we must note that the bound is only valid for instances which satisfy the triangle-inequality, because the Christofides bound is only valid for those instances.

**Theorem 2.2.5.** *For all instances  $I$  of  $FAP1(\delta)$  which satisfy the triangle-inequality,  $\frac{Z_\delta^{NAT-C}(I)}{Z_\delta^*(I)} < \frac{5}{2} - \frac{1}{\delta}$ .*

*Proof.*

$$\begin{aligned} \max_I \frac{c_\delta(T_{1C})}{c_\delta(T_\delta)} &\leq \frac{c_1(T_{1C}) + (n-1)(\delta-1)}{c_\delta(T_\delta)} \leq \frac{\frac{3}{2}c_1(T_\delta) + (n-1)(\delta-1)}{c_\delta(T_\delta)} \\ &\leq \frac{\frac{3}{2}c_\delta(T_\delta) + (n-1)(\delta-1)}{c_\delta(T_\delta)} < \frac{3}{2} + \frac{(n-1)(\delta-1)}{(n-1)\delta} = \frac{5}{2} - \frac{1}{\delta}. \end{aligned} \quad (18)$$

□

For heuristic  $NAT - C$ , we do not have an example showing the worst-case performance bound is tight.

In Section 2.2.2, we developed heuristics which use TSP solution methods as subroutines and then transform the resulting tours to depot-based appointment time tours. These heuristics produce feasible solutions to  $FAP1(\delta)$  but are not guaranteed to produce optimal solutions. In the next section, we develop network transformations that enable the use of TSP solution methods for solving  $FAP1(\delta)$ .

### 2.2.3 Network transformations for $FAP1(\delta)$

Some  $FAP1(\delta)$  instances can be solved via traditional TSP methods when an appropriate network transformation is used. Below, we discuss two transformations and how they can be used.

#### 2.2.3.1 Equal service time customers, appointment time constraints at depot

In this section, we consider the set of instances where:

- the service times at all customers are equal, i.e.,  $w_i = w \ \forall i \in \mathcal{N}$ , and
- the vehicle is restricted to leave and return to the depot at an appointment time.

Although we are primarily interested in the case in which the appointment time restrictions are not present at the depot, we continue with the assumption throughout Section 2.2.3.1 to establish basic properties of the transformed network.

The network transformation is as follows. Create  $\mathcal{G}' = (\mathcal{N}, \mathcal{A})$  from the original network  $\mathcal{G}$  by replacing arc costs  $h_{ij}$  as shown in Equation (19):

$$\bar{h}_{ij} = \left\lceil \frac{h_{ij} + w}{\delta} \right\rceil \delta. \quad (19)$$

First note that because  $\mathcal{G}$  is undirected, the transformed network in this case can remain undirected because  $h_{ij} \equiv h_{ji}$ . Using Lemma 2.2.2, we can show that  $\mathcal{G}'$  satisfies the triangle-inequality.

**Lemma 2.2.2.** *The ceiling function is subadditive.*

**Theorem 2.2.6.** *The arc costs  $\bar{h}_{ij}$  in  $\mathcal{G}'$  as defined in Equation (19) satisfy the triangle-inequality if:*

- *the arc costs  $h_{ij}$  on the undirected symmetric network  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$  satisfy the triangle-inequality, and*
- *$w_i = w \geq 0 \ \forall i \in \mathcal{N}$ .*

*Proof.* From the triangle-inequality on  $\mathcal{G}$ , we have  $h_{ik} \leq h_{ij} + h_{jk} \ \forall i, j, k \in \mathcal{A}$ , therefore we also have  $\frac{h_{ik}+w}{\delta} \leq \frac{h_{ij}+w}{\delta} + \frac{h_{jk}+w}{\delta}$ , and  $\left\lceil \frac{h_{ik}+w}{\delta} \right\rceil \leq \left\lceil \frac{h_{ij}+w+h_{jk}+w}{\delta} \right\rceil$ . Using Lemma 2.2.2 and multiplying by  $\delta$ , we have that  $\left\lceil \frac{h_{ik}+w}{\delta} \right\rceil \delta \leq \left\lceil \frac{h_{ij}+w+h_{jk}+w}{\delta} \right\rceil \delta \leq \left\lceil \frac{h_{ij}+w}{\delta} \right\rceil \delta + \left\lceil \frac{h_{jk}+w}{\delta} \right\rceil \delta$ .  $\square$

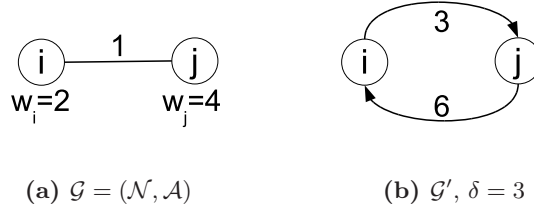
We have established that the transformed network  $\mathcal{G}'$  with arc costs defined in Equation (19) satisfies the triangle-inequality if  $\mathcal{G}$  satisfied the triangle-inequality, the service times at all customers are equal, and the vehicle is constrained to leave and return to the depot at an appointment time. In the problem setting of primary interest, we do not include a constraint that the vehicle must leave and return to the depot at an appointment time. Additionally, the service times at all customers need not necessarily be equal.

### 2.2.3.2 Unequal service time customers, no appointment time constraints at the depot

The impact of removing appointment time constraints at the depot is that, in any tour  $T$  that begins and ends at the depot, two fewer travel segments can induce idle time into the total tour duration. Thus, consider the following network transformation, which replaces the arc costs in  $\mathcal{G}'$  with those shown in Equation (20):

$$\bar{h}_{ij} = \begin{cases} h_{ij} & \text{if } i = 0 \text{ or } j = 0, \\ \left\lceil \frac{h_{ij} + w_i}{\delta} \right\rceil \delta & \text{otherwise.} \end{cases} \quad (20)$$

This transformation does not preserve the undirected and symmetric properties of the network, as illustrated in Figures 4a and 4b, which show a network both before and after the transformation.



**Figure 4:** Asymmetric and undirected transformed network

Solving the appointment time problem on  $\mathcal{G}$  requires solving an asymmetric TSP on  $\mathcal{G}'$ . We are able to show that solving an asymmetric TSP optimally on  $\mathcal{G}'$  solves FAP1( $\delta$ ) optimally on  $\mathcal{G}$ .

**Theorem 2.2.7.** *When the service times at customers are not equal, and the vehicle is not constrained to leave and return to the depot at appointment times, the optimal appointment time tour  $T_\delta$  on  $\mathcal{G}$  can be found by solving an asymmetric TSP on  $\mathcal{G}'$ .*

*Proof.* Let  $T'$  be the tour found by solving the asymmetric TSP on  $\mathcal{G}'$  with arc costs  $\bar{h}_{ij}$ . We know that:

$$\sum_{(i,j) \in T'} \bar{h}_{ij} \leq \sum_{(i,j) \in T} \bar{h}_{ij} \quad \forall T \in \mathcal{G}'. \quad (21)$$



Let  $T_\delta$  be the optimal appointment time tour on  $\mathcal{G}$  with arc costs  $h_{ij}$ , and let  $c_\delta(T_\delta)$  be the duration of the tour under appointment times, calculated as described in Equation (3).

We know that:

$$c_\delta(T_\delta) = h_{01} + \delta \sum_{i=1}^{n-1} \left\lceil \frac{h_{i,i+1} + w_i}{\delta} \right\rceil + w_n + h_{n0} \leq c_\delta(T), \quad \forall T \in \mathcal{G}. \quad (22)$$

Now, suppose  $T'$  is not the optimal appointment time tour on  $\mathcal{G}$ . Then,

$$\begin{aligned} c_\delta(T') &= h_{01} + \delta \sum_{i=1}^{n-1} \left\lceil \frac{h_{i,i+1} + w_i}{\delta} \right\rceil + w_n + h_{n0} \\ &> h_{01} + \delta \sum_{i=1}^{n-1} \left\lceil \frac{h_{i,i+1} + w_i}{\delta} \right\rceil + w_n + h_{n0} = c_\delta(T_\delta). \end{aligned} \quad (23)$$

It would immediately follow that:

$$\sum_{(i,j) \in T'} \bar{h}_{ij} > \sum_{(i,j) \in T_\delta} \bar{h}_{ij}. \quad (24)$$

Any set of arcs that make up a tour  $\mathcal{G}$  also make up a tour in  $\mathcal{G}'$ . Hence, the arcs making up  $T_\delta$  in  $\mathcal{G}$  would create a tour in  $\mathcal{G}'$  with lower cost than  $T'$ , and  $T'$  could not be the optimal TSP tour in  $\mathcal{G}'$ .  $\square$

### 2.2.3.3 Concluding remarks regarding FAP1( $\delta$ )

We have seen that offering fixed appointment times for customer visits can increase the duration of tours which visit all customers. When  $\delta = 1$  and all problem data is integral, FAP1( $\delta$ ) is equivalent to the TSP. Before concluding the discussion of FAP1( $\delta$ ), we point out some additional special instances of FAP1( $\delta$ ).

1. For every pair of customer locations  $(i, j)$  in  $\mathcal{N}$ , there exists some positive integer  $k$  such that  $h_{ij} + w_i = k\delta$ . Additionally, for every arc  $(0, i)$  between the depot and a customer location, there exists some positive integer  $l$  such that  $h_{0i} = l\delta$ .

In this case, solving the asymmetric TSP using the original arc cost data solves FAP1( $\delta$ ) because idle time is never induced on any arc, and thus each transformed arc cost never differs from the original.

2. For some integer  $k \geq 0$ ,  $k\delta \leq h_{ij} + w_i \leq (k+1)\delta \forall (i, j) \in \mathcal{A}$ .

In the transformed network, every arc between two customer locations in the network will have cost  $(k+1)\delta$ . Any visit sequence which minimizes  $h_{01} + n_{n0}$  will minimize the duration of the depot-based appointment time tour.

### 2.3 *FAP2*( $\delta, T_{max}$ ): *maximize number of customers served given tour duration constraint*

In the preceding section, we assumed that a sufficient number of appointment times were available to allow serving all customers, and the objective we considered was minimizing the duration of a tour that served all customers. In most practical applications, tour duration is not unconstrained. It is often limited by the length of a workday, the number of available appointment times, the capacity of a vehicle, or some other limiting resource.

In this section, we study the impact fixed appointment times have on the number of customers that can be visited if tour duration is limited. We begin by stating the problem and commenting on its complexity. Next, we develop a bound on the ratio of the number of customers that can be served when fixed appointment times are used versus when they are not for a special case of the problem. We consider the effectiveness of traditional distance-minimizing heuristics for this class of problems, and provide a bound for their worst-case performance. We also analyze the appropriateness of the network transformation discussed in the preceding section.

#### 2.3.1 Problem statement

There is a set of  $\mathcal{N}$  customer locations, with each customer  $i \in \mathcal{N}$  requiring service time  $w_i \geq 0$ . All customer requests are known in advance. A complete set of undirected arcs  $(i, j) \in \mathcal{A}$  connects all customer locations. The travel time  $h_{ij} \geq 0$  on each arc is known and the network  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$  satisfies the triangle-inequality. One vehicle is available to serve customers using a menu of equally-spaced appointment times  $\{a + k\delta : k = 0, \dots, m\}$ , where  $a$  is the earliest appointment time,  $m+1$  is the total number of appointments, and  $\delta$  is the time elapsed between each allowable appointment. Let  $q_i$  denote the appointment time assigned to customer  $i$ . The vehicle leaves the depot at time  $q_F - h_{0F}$ , in time to travel to the

first customer appointment. The vehicle returns to the depot at time  $q_L + w_{i_L} + h_{L0}$ , after serving the last customer appointment. The total tour duration is constrained by an upper bound,  $T_{max}$ . The problem is to assign appointment times  $q_i$  from the appointment time menu to the maximum number of customers served, such that the duration of the resulting tour serving those customers is minimized and is less than  $T_{max}$ . Note that  $\text{FAP1}(\delta) = \text{FAP2}(\delta, \infty)$ .

Let  $\mathcal{N}_p$  denote a subset of  $\mathcal{N}$  of size  $p$ , for  $p \leq |\mathcal{N}|$ . Solving  $\text{FAP2}(\delta, T_{max})$  requires answering the question: what is the largest subset  $\mathcal{N}_p \subseteq \mathcal{N}$  that can be found, for which the minimum duration depot-based appointment time tour with parameter  $\delta$  serving all customers in  $\mathcal{N}_p$  has duration less than or equal to  $T_{max}$ ? We can solve an instance of  $\text{FAP2}(\delta, T_{max})$  through a series of repeated calls to  $\text{FAP1}(\delta)$ , as shown in Algorithm 2. Suppose  $P$  is a valid upper bound on the number of customers that can be served for a given instance of the problem. First, an instance of  $\text{FAP1}(\delta)$  is solved for all possible subsets of size  $P$ . If the minimum duration tour across all such subsets has duration less than  $T_{max}$ , a solution has been obtained to  $\text{FAP2}(\delta, T_{max})$ . Otherwise, iteratively consider subsets of smaller size.

---

**Algorithm 2** Algorithm for  $\text{FAP2}(\delta, T_{max})$

---

```

1:  $p = P$ 
2: for all  $\mathcal{N}_p \subseteq \mathcal{N}$  do
3:   Solve  $\text{FAP1}(\delta)$  on subgraph  $\mathcal{G}_p$  induced by  $\mathcal{N}_p$ ; let  $T$  be tour returned
4:   Add  $T$  to the set  $\mathcal{T}_p$ 
5: end for
6: Let  $T^* = \arg \min \{c_\delta(T) : T \in \mathcal{T}_p\}$ 
7: if  $c_\delta(T^*) \leq T_{max}$  then
8:   return  $T^*$ 
9: else
10:  Set  $p = p - 1$  and return to step 2
11: end if

```

---

No tour can visit more customers than are in the network, hence  $P \leq |\mathcal{N}|$ . Given integral problem data, a tour visiting  $P$  customers plus the depot must have duration at least  $P + 1$ , hence  $P + 1 \leq T_{max}$ . A selection for  $P$  that is valid for all instances of  $\text{FAP2}(\delta, T_{max})$  is defined in Equation (25):

$$P \leq \min(|\mathcal{N}|, T_{max} - 1). \quad (25)$$

We focus the following analysis on problem instances where tour duration, not the number of customers in the network, is the limiting constraint. Otherwise, the problem reduces to FAP1( $\delta$ ) when the minimum duration tour through  $|\mathcal{N}|$  customers has duration less than  $T_{max}$ .

### 2.3.2 Complexity of FAP2( $\delta, T_{max}$ )

FAP2( $\delta, T_{max}$ ) is in a set of optimization problems classified by Feillet et al. [22] as Traveling Salesman Problems with Profits (TSPs with profits). In such problems,  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  is a graph where  $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$  is a set of vertices, with  $v_1$  being the depot, and  $\mathcal{E}$  is a set of edges. There is a profit  $p_i$  associated with each vertex other than the depot, and a distance  $c_{ij}$  associated with each edge. TSPs with profits seek to balance the costs and profits associated with tours which visit each vertex at most once and include the depot. The specific problem variant within TSPs with Profits most similar to FAP2( $\delta, T_{max}$ ) is the Orienteering Problem (OP). In the OP, the objective is to maximize the profit collected while the resultant tour does not exceed a constraint on maximum travel, and minimize the duration of the tour used to visit the maximum number of customers.

FAP2( $\delta, T_{max}$ ) can be modeled as an OP. Let  $T_{max}$  be the constraint on the maximum duration of the tour. Let the profit of each vertex, excluding the depot, be one. Let the cost on each arc include service time, travel time, and idle time, as defined in Equation (20). As discussed in Section 2.2.3, this transformation could yield a network that is directed and asymmetric. Solving the asymmetric OP with this definition of problem parameters will solve the instance of FAP2( $\delta, T_{max}$ ). Additional details on this topic are given in Section 2.3.6.

Laporte and Martello [34] shows that the OP, referred to as the selective traveling salesman problem (STSP), remains NP-hard for the equal-profits case. Therefore, FAP2( $\delta, T_{max}$ ) is NP-hard because OP with equal profits is a special case of FAP2( $\delta, T_{max}$ ) where each customer location has zero service time and the appointment time grid spacing parameter  $\delta$

is equal to one. While  $\text{FAP2}(\delta, T_{max})$  is NP-hard, there are certain classes of instances for which the solution is trivial. These are discussed next.

### 2.3.2.1 Easy instances of $\text{FAP2}(\delta, T_{max})$

1.  $w_i + h_{ij} \leq \delta \forall i, j$

In this case, customers can be visited in any order without affecting tour duration. Customer  $i + 1$  immediately preceded by customer  $i$  with appointment time  $q_i$  will have appointment time  $q_{i+1} = q_i + \delta$ . The maximum number of customers that can be visited is  $\lfloor \frac{T_{max}-2}{\delta} \rfloor$ . The minimum duration required to do so is equal to  $(n - 1)\delta + 2$ . The specific customers which are included in the subset to serve may be selected arbitrarily.

2.  $w_i = w \forall i$  and  $h_{ij} = h \forall (i, j)$

In this case, customers can again be visited in any order without affecting tour duration. If  $w + h \leq \delta$ , case (i) results. If  $w + h > \delta$ , customer  $i + 1$  immediately preceded by customer  $i$  with appointment time  $q_i$  will have appointment time  $q_{i+1} = q_i + k\delta$ , where  $k = \lceil \frac{w+h}{\delta} \rceil$ . The maximum number of customers that can be visited is  $\lfloor \frac{T_{max}-2}{k\delta} \rfloor$  and the minimum duration required to do so is equal to  $(n - 1)k\delta + 2$ . The specific customers which are included in the subset to serve may be selected arbitrarily.

### 2.3.3 Impact of appointment times on number of customers visited

Because the primary objective of  $\text{FAP2}(\delta, T_{max})$  is to maximize the number of customers that can be served, we wish to compare the number of customers that can be visited when fixed appointment times are used versus when they are not.

Recall that under integral problem data, setting the grid spacing parameter  $\delta$  equal to 1 is equivalent to the unrestricted case. Suppose we have a tour  $T_1$  which visits  $n_1$  customers in  $\mathcal{N}$ . The duration of the tour,  $c_1(T_1)$  as defined in Equation (5), will only include visit time and travel time. Suppose that for this instance,  $T_{max} = c_1(T_1)$ , hence  $T_1$  represents a feasible solution to  $\text{FAP2}(1, T_{max})$ . Suppose also that:

- there is no tour  $T' \in \mathcal{G}$  which visits more than  $n_1$  customers, and

- there is no tour  $T' \in \mathcal{G}$  which visits exactly  $n_1$  customers with duration  $c_1(T') < c_1(T_1)$ .

Then,  $T_1$  must be an optimal solution to this instance of FAP2(1,  $T_{max}$ ). We wish to understand: if an appointment time menu with parameter  $\delta > 1$  is implemented, how will that impact the number of customers it is possible to visit? Let the number of customers served in an optimal solution to FAP2( $\delta, T_{max}$ ) be  $n_\delta$ . We first focus attention on an instance for which a bound on the ratio  $\frac{n_1}{n_\delta}$  can be proven.

### 2.3.3.1 Bound on $\frac{n_1}{n_\delta}$ for special set of instances

Consider the set of FAP2( $\delta, T_{max}$ ) instances with service time  $w = 0$  at all customers and travel time  $h = 1$  on all arcs. We assume that  $T_{max} \geq 2$ , which is the minimum duration required to serve at least one customer. We have already identified instances of this type as easy. An arbitrary subset of customers can be assigned appointment times in an arbitrary order, as long as each appointment begins at the earliest feasible time and the tour duration constraint is not violated. We prove a bound of  $1 \leq \frac{n_1}{n_\delta} \leq \delta$  for instances of this type in Theorem 2.3.1. In order to prove the bound on the ratio  $\frac{n_1}{n_\delta}$ , we need a valid lower bound on  $n_\delta$ . Thus, we first establish Lemma 2.3.1.

**Lemma 2.3.1.** *On instances of FAP2( $\delta \geq 2, T_{max}$ ) where  $h_{ij} = h = 1 \forall (i, j) \in \mathcal{A}$ ,  $w_i = w = 0 \forall i \in \mathcal{N}$ , and  $T_{max} \geq 2$ ,  $n_\delta \geq \frac{1}{\delta}(T_{max} - 1)$ .*

*Proof.* We will consider two cases:

1.  $\frac{1}{\delta}(T_{max} - 1) \in \mathbb{Z}^+$
2.  $\frac{1}{\delta}(T_{max} - 1) \notin \mathbb{Z}^+$

For case 1, where  $\frac{1}{\delta}(T_{max} - 1) \in \mathbb{Z}^+$ , we will prove the claim  $n_\delta \geq \frac{1}{\delta}(T_{max} - 1)$  by contradiction. Suppose instead that  $n_\delta < \frac{1}{\delta}(T_{max} - 1) \in \mathbb{Z}^+$ . We will show this cannot be true. Finding the optimal  $n_\delta$  for FAP2( $\delta, T_{max}$ ) requires solving the following integer program:

$$\text{Max } n_\delta \quad (26)$$

$$\text{s.t. } (n_\delta - 1)\delta + 2 \leq T_{max} \quad (27)$$

$$n_\delta \in \mathbb{Z}^+ \quad (28)$$

The optimal solution  $n_\delta$  to this integer program is:

$$n_\delta = \begin{cases} 1 + \frac{T_{max}-2}{\delta} & \text{if } \frac{T_{max}-2}{\delta} \in \mathbb{Z}^+ \\ 1 + \lceil \frac{T_{max}-2}{\delta} \rceil - 1 & \text{otherwise.} \end{cases} \quad (29)$$

When  $\delta \geq 2$ , we cannot have  $\frac{T_{max}-1}{\delta} \in \mathbb{Z}^+$  and  $\frac{T_{max}-2}{\delta} \in \mathbb{Z}^+$  simultaneously. Thus, the optimal  $n_\delta$  must be equal to  $\lceil \frac{T_{max}-2}{\delta} \rceil$  according to Equation (29). For  $\delta \geq 2$  and  $\frac{1}{\delta}(T_{max} - 1) \in \mathbb{Z}^+$ , definition of the ceiling function requires that  $\lceil \frac{T_{max}-2}{\delta} \rceil = \frac{1}{\delta}(T_{max} - 1)$ . This provides the contradiction, because  $n_\delta = \lceil \frac{T_{max}-2}{\delta} \rceil = \frac{1}{\delta}(T_{max} - 1) \not\in \mathbb{Z}^+$ . Thus,  $n_\delta \geq \frac{1}{\delta}(T_{max} - 1)$  for case 1.

Consider now case 2, where  $\frac{1}{\delta}(T_{max} - 1) \notin \mathbb{Z}^+$ . We know from Equation (29) that  $n_\delta$  is equal to one of two values dependent on whether  $\frac{T_{max}-2}{\delta} \in \mathbb{Z}^+$ . Thus, we again have two cases to consider, for which proofs that  $n_\delta > \frac{T_{max}-1}{\delta} \notin \mathbb{Z}^+$  are given below.

$$1. \frac{T_{max}-2}{\delta} \in \mathbb{Z}^+$$

There exists some  $k \in \mathbb{Z}^+ = \frac{T_{max}-2}{\delta}$ , so  $n_\delta = k + 1$ . Hence, for  $\delta \geq 2$ ,

$$k < \frac{T_{max} - 1}{\delta} < k + 1 = n_\delta. \quad (30)$$

$$2. \frac{T_{max}-2}{\delta} \notin \mathbb{Z}^+$$

For  $\delta \geq 2$ , if neither  $\frac{T_{max}-2}{\delta} \in \mathbb{Z}^+$  nor  $\frac{T_{max}-1}{\delta} \in \mathbb{Z}^+$ , then:

$$n_\delta = \left\lceil \frac{T_{max} - 2}{\delta} \right\rceil = \left\lceil \frac{T_{max} - 1}{\delta} \right\rceil > \frac{T_{max} - 1}{\delta}. \quad (31)$$

□

Having established Lemma 2.3.1, we can prove Theorem 2.3.1.

**Theorem 2.3.1.** *For an instance of  $FAP2(\delta, T_{max})$  with  $h_{ij} = h = 1 \forall (i, j) \in \mathcal{A}$ ,  $w_i = 0 \forall i \in \mathcal{N}$ , and  $T_{max} \geq 2$ ,  $1 \leq \frac{n_1}{n_\delta} \leq \delta$ .*

*Proof.* It should be clear that  $n_\delta$  can never be greater than  $n_1$  because a depot-based appointment time tour using parameter  $\delta > 1$  may include idle time in addition to travel and visit time. If there exists a tour  $T_\delta$  serving  $n_\delta$  customers with duration  $c_\delta(T_\delta) \leq T_{max}$ , then that same sequence of customer visits is also feasible when  $\delta = 1$  because  $c_1(T_\delta) \leq c_\delta(T_\delta)$ .

To establish the upper bound on  $\frac{n_1}{n_\delta}$ , note that we have a lower bound on the denominator from Lemma 2.3.1, and we can easily show an upper bound of  $T_{max} - 1$  on the numerator. On this instance, it is clear that  $n_1 = T_{max} - 1$ . Visiting the depot plus  $n_1$  customers when all  $h_{ij} = 1$ , all  $w_i = 0$ , and  $\delta = 1$  requires  $n_1 + 1$  total units of time. Thus, the relationship shown in Equation (32) is established, proving the theorem:

$$\frac{n_1}{n_\delta} \leq \frac{T_{max} - 1}{\frac{1}{\delta}(T_{max} - 1)} \leq \delta. \quad (32)$$

□

An example showing this bound is tight is given in the next section.

### 2.3.3.2 Tight example for Theorem 2.3.1

In the previous section, we established that  $1 \leq \frac{n_1}{n_\delta} \leq \delta$  for a subset of  $FAP2(\delta, T_{max})$  instances having all travel times equal to 1 and all service times equal to 0. For an example showing the lower bound is tight, let  $T_{max} = 2$ . Then, both  $T_1$  and  $T_\delta$  serve just one customer plus the depot, and  $\frac{n_1}{n_\delta} = 1$ .

Generating an example where the upper bound of  $\delta$  on the ratio is tight is also easy. Consider a network with 3 nodes plus the depot, and suppose  $T_{max} = 4$  and  $\delta = 3$ . When  $\delta = 1$ , it is possible to visit all three customers plus the depot with a total tour duration of 4. When  $\delta = 3$ , the appointment time tour can visit at most one customer plus the depot with a tour duration of 2. Visiting an additional customer would increase the tour duration to 5. In this example,  $\frac{n_1}{n_\delta} = \frac{3}{1} = 3 = \delta$ .



**Table 1:** Arc costs for  $\mathcal{G}$ 

	Depot	A	B	C	D	E	F	G	H	I
Depot	0	8	4	2	2	8	3	8	7	2
A	8	0	6	6	6	2	11	16	15	10
B	4	6	0	2	6	6	7	12	11	6
C	2	6	2	0	4	6	5	10	9	4
D	2	6	6	4	0	6	5	10	9	4
E	8	2	6	6	6	0	11	16	15	10
F	3	11	7	4	5	11	0	5	10	5
G	8	16	12	10	10	16	5	0	5	10
H	7	15	11	5	9	15	10	5	0	5
I	2	10	6	10	4	10	5	10	5	0

### 2.3.3.3 Bound on $\frac{n_1}{n_\delta}$ for other problem instances

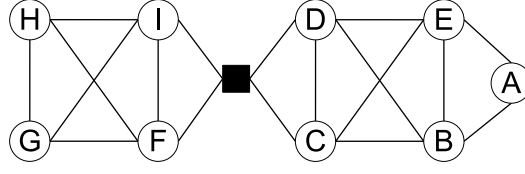
We established a provable upper bound of  $\delta$  on the ratio of the optimal number customers served when appointment times are not used versus when they are for problem instances where all service times are equal to some  $w > 0$  and all arc costs are equal to some  $h$ , and  $T_{max} \geq 2h$ . However, when service times and travel times are not homogenous, we are not able to establish meaningful bounds on  $\frac{n_1}{n_\delta}$  which are provable across all instances. The difficulty arises from the fact that the subsets of customers selected in optimal solutions to  $\text{FAP2}(1, T_{max})$  and  $\text{FAP2}(\delta > 1, T_{max})$  may be different. The subgraphs induced by those subsets are therefore difficult to compare.

### 2.3.4 Customer subsets selected in optimal solutions to $\text{FAP2}(\delta, T_{max})$

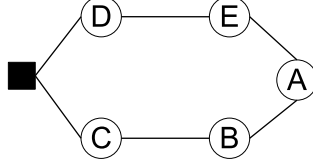
In this section, we wish to understand the relationship between the subsets of customers selected in optimal solutions to  $\text{FAP2}(1, T_{max})$  and  $\text{FAP2}(\delta > 1, T_{max})$ . Let  $\mathcal{N}_1$  and  $\mathcal{N}_\delta$  denote those subsets, respectively. Define subgraphs  $\mathcal{G}_1$  and  $\mathcal{G}_\delta$ , which contain only those customer locations in  $\mathcal{N}_1$  and  $\mathcal{N}_\delta$ , respectively. The arc set in each subgraph includes only those arcs for which both endpoints are in the respective set of customer locations.

It is clear that the relationship  $|\mathcal{N}_\delta| \leq |\mathcal{N}_1|$  must hold. However, it is not necessary that  $\mathcal{N}_\delta$  must be a subset of  $\mathcal{N}_1$ . Consider the example network shown in Figure 5a with arc costs given in Table 1. Let  $T_{max} = 20$  and  $\delta = 5$ .

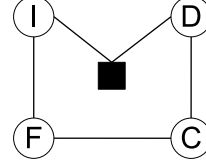
An optimal solution to  $\text{FAP2}(1, 20)$  is shown in Figure 5b, and an optimal solution to  $\text{FAP2}(5, 20)$  is shown in Figure 5c. In these solutions,  $n_\delta = 4$ ,  $\mathcal{N}_\delta = \{C, D, F, I\}$ ,  $n_1 = 5$ , and  $\mathcal{N}_1 = \{A, B, C, D, E\}$ . As expected,  $n_\delta \leq n_1$ , but  $\mathcal{N}_\delta \not\subseteq \mathcal{N}_1$ .



(a)  $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ ,  $T_{max} = 20$ ,  $\delta = 5$



(b)  $n_1 = 5$ ,  $c_1(T) = 20$ ,  $c_\delta(T) = 34$



(c)  $n_\delta = 4$ ,  $c_1(T) = 18$ ,  $c_\delta(T) = 19$

**Figure 5:** Example network,  $T_{max} = 20$ ,  $\delta = 5$

This observation has a significant impact on the heuristics and algorithms that can be used to solve  $\text{FAP2}(\delta, T_{max})$ . Suppose, for example, we use a heuristic which first solves  $\text{FAP2}(1, T_{max})$ , then solves  $\text{FAP2}(\delta > 1, T_{max})$  on the subgraph  $\mathcal{G}_1$ . In this example, the second step requires solving  $\text{FAP2}(5, 20)$  on  $\mathcal{G}_1$  which includes customer locations  $\{A, B, C, D, E\}$ . It is not possible to find a depot-based appointment time tour which serves all five customers in  $\mathcal{N}_1$  with duration less than  $T_{max} = 20$ . It is also not possible to find an appointment time tour serving four customers from  $\mathcal{N}_1$ . The minimum duration appointment time tour on all possible subsets of  $\mathcal{N}_1$  of size four is  $T = \{0-C-B-E-D-0\}$  with duration  $c_5(T) = 29$ . The heuristic clearly then does not optimally solve this instance. In the next section, we consider a TSP-based heuristic for  $\text{FAP2}(\delta, T_{max})$  which uses heuristic  $NAT$  for  $\text{FAP1}(\delta)$  as a subroutine.

### 2.3.5 TSP-based solution method for $\text{FAP2}(\delta, T_{max})$

The heuristic presented in this section differs from the heuristic in the previous section in that it inspects candidate TSP tours on various-sized subsets for appointment time duration feasibility throughout the solution procedure. The heuristic uses  $NAT$  as a subroutine. Recall that  $NAT$  solves an instance of  $\text{FAP1}(\delta)$  by first solving a TSP on  $\mathcal{G}$  and then

transforming the tour to a depot-based appointment time tour. Also recall that *NAT* assumes we have access to the *optimal* TSP solution, but it is not guaranteed to produce *optimal* solutions to FAP1( $\delta$ ). Heuristic *NAT2*, described below, solves FAP2( $\delta, T_{max}$ ) by finding the largest sized subset  $\mathcal{N}_i$  for which the tour  $T_\delta$  returned by *NAT* on the subgraph  $\mathcal{G}_i$  has duration  $c_\delta(T_\delta) \leq T_{max}$ .

Note that *NAT2* has an enumeration feature which will be very slow when  $P$  is large. In each iteration of the heuristic corresponding to some value of  $P$ , *NAT* is called  $\binom{n}{P}$  times; once for every possible subset of size  $P$  in  $\mathcal{N}$ . Binary search ideas may be utilized to make the process faster. Instead of setting  $P$  to some maximum value in the first iteration and decrementing  $P$  by 1 in subsequent iterations, a modified heuristic could be implemented by placing  $1, \dots, T_{max}$  (or the values between some feasible lower and upper bound on customers served) in a sorted list. An initial value for  $P$  would be selected from the middle of the list. If a tour through any subset of  $P$  customer locations had duration less than  $T_{max}$ , the lower half of the list would be eliminated from consideration; otherwise, the upper half would be eliminated. The next value for  $P$  would be selected from the middle of the remaining list.

---

**Algorithm 3** *NAT2*: TSP-based approach for FAP2( $\delta, T_{max}$ )

---

```

1: Set  $P = T_{max} - 2 \min\{c_{0i} : (0, i) \in \mathcal{A}\}$ 
2: for all  $\mathcal{N}_P \subseteq \mathcal{N}$  do
3:   Use NAT to solve FAP1( $\delta$ ) on subgraph of  $\mathcal{G}$  induced by  $\mathcal{N}_P$ ; let  $T_\delta$  be tour returned
4:   Add  $T_\delta$  to the set  $\mathcal{T}_P$ 
5: end for
6: Let  $T_\delta^* = \arg \min\{c_\delta(T_\delta) : T_\delta \in \mathcal{T}_P\}$ 
7: if  $c_\delta(T_\delta^*) \leq T_{max}$  then
8:   return  $T_\delta^*$ 
9: else
10:  Set  $P = P - 1$  and return to step 2
11: end if

```

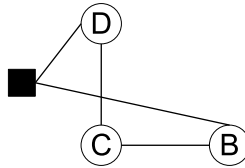
---

A feasible solution to FAP2( $\delta, T_{max}$ ) specifies both a set of customers to be visited, and a depot-based appointment time tour serving those customers having duration less than  $T_{max}$ . Step 7 of *NAT2* guarantees the duration of the tour returned is within the allowable limit. Thus, *NAT2* returns feasible solutions.

For  $\delta > 1$ , *NAT2* is not guaranteed to return optimal solutions to  $\text{FAP2}(\delta, T_{\max})$ . By Theorem 2.2.3, *NAT* is not guaranteed to produce optimal solutions to  $\text{FAP1}(\delta)$ . Thus, the tour specified by *NAT2* for the selected set of customers may not be optimal. Furthermore, it may be possible to find a larger subset of customers to visit.

To see this, consider again the network example in Figure 5a with arc costs given in Table 1 and parameters  $(\delta, T_{\max}) = (5, 20)$ . The optimal solution to  $\text{FAP1}(1, 20)$  serves 5 customers, as shown in Figure 5b. Because *NAT2* cannot obtain a solution serving more than 5 customers if  $\delta > 1$ , we begin with  $P = 5$  and show the final three iterations of *NAT2* below. The resultant tour serving 3 customers is given in Figure 6. The optimal solution to this instance of  $\text{FAP2}(5, 20)$  serves 4 customers (Figure 5c).

- $P = 5$ :  $T_{\delta}^* = \{0-D-E-A-B-C-0\}$ ,  $c_{\delta}(T_{\delta}^*) = 34 > 20$
- $P = 4$ :  $T_{\delta}^* = \{0-C-B-E-D-0\}$ ,  $c_{\delta}(T_{\delta}^*) = 29 > 20$
- $P = 3$ :  $T_{\delta}^* = \{0-D-C-B-0\}$ ,  $c_{\delta}(T_{\delta}^*) = 16 \leq 20$



**Figure 6:** Counterexample for *NAT2*,  $n_5 = 3$ ,  $c_1(T) = 12$ ,  $c_5(T) = 16$

The result that *NAT2* is not guaranteed to produce optimal solutions is not surprising. As seen in Section 2.2.2, using TSP-based heuristics to find appointment time tours can produce sub-optimal results unless a transformed network is used. In the next section, we present a network transformation that enables the use of OP heuristics in solving instances of  $\text{FAP2}(\delta, T_{\max})$ .

### 2.3.6 Network transformation for $\text{FAP2}(\delta, T_{\max})$

Recall that  $\text{FAP2}(\delta, T_{\max})$  can be modeled as an OP. In this section, we present a network transformation that can be used in combination with asymmetric OP solution methods to

solve instances of  $\text{FAP2}(\delta, T_{max})$ . The network transformation is as follows.

**Network transformation:** Create  $\mathcal{G}' = (\mathcal{N}, \mathcal{A})$  from the original network  $\mathcal{G}$  by replacing arc costs  $h_{ij}$  as determined by Equation (20). Also, introduce a profit  $p_i = 1$  for each customer location  $i \in \mathcal{N}$ .

This transformation yields a network which may be undirected and asymmetric. We can show that solving the asymmetric OP on  $\mathcal{G}'$  solves  $\text{FAP2}(\delta, T_{max})$  on  $\mathcal{G}$ . The theorem and proof follow.

**Theorem 2.3.2.** *Solving the asymmetric OP on  $\mathcal{G}'$  created using the network transformation described in Section 2.3.6 solves  $\text{FAP2}(\delta, T_{max})$  on  $\mathcal{G}$ .*

*Proof.* 1. Let  $T_{OP}$  be the optimal solution to the OP on  $\mathcal{G}'$ , serving  $n_{OP}$  customers and having total cost  $\sum_{(i,j) \in T_{OP}} \bar{h}_{ij} \leq T_{max}$ . If this tour exists in  $\mathcal{G}'$ , it also exists in  $\mathcal{G}$ , and  $c_\delta(T_{OP}) \leq T_{max}$ . Thus,  $T_{OP}$  is a depot-based appointment time tour in  $\mathcal{G}$  serving  $n_{OP}$  customers having duration less than the allowable limit. Furthermore,  $T_{OP}$  is the minimum duration tour in  $\mathcal{G}$  serving that particular subset of  $n_{OP}$  customers. This follows from Theorem 2.2.7.

2. There does not exist a tour  $T$  in  $\mathcal{G}$  that serves  $m > n_{OP}$  customers with duration  $c_\delta(T) < T_{max}$ . Otherwise, there would have been a tour  $T' \in \mathcal{G}'$  visiting  $m$  customers with total cost  $\sum_{(i,j) \in T'} \bar{h}_{ij} \leq T_{max}$ . Then,  $T_{OP}$  would not have been the optimal solution to the OP on  $\mathcal{G}'$  because  $T'$  serves  $m > n_\delta$  customers. When the profit at all customers is set to 1, the objective of the OP is to maximize the number of customers served, and minimize the duration of the tour used to serve the maximum number of customers.

3. There does not exist a tour in  $\mathcal{G}$  that serves a different subset of  $n_\delta$  customers with lower appointment time duration than  $c_\delta(T_{OP})$ . Let  $n^*$  be the number of customers served in an optimal solution to OP. The OP also returns the minimum duration tour over all possible subsets of size  $n^*$ . Thus,  $T_{OP}$  must have been the minimum duration tour over all possible subsets of size  $n_{OP} \in \mathcal{G}'$ . And again, by Theorem 2.2.7, that

must mean  $T_{OP}$  is the tour with minimum appointment time duration  $c_\delta$  over all possible tours serving subsets of customers of size  $n_{OP} \in \mathcal{G}$ .

□

### 2.3.7 Concluding remarks regarding $\text{FAP2}(\delta, T_{max})$

In Section 2.3, it was shown that offering fixed appointment times for customer visits can decrease the number of customers that can be served given an upper limit on tour duration. When  $\delta = 1$  and the profit at all customers is set to 1,  $\text{FAP2}(\delta, T_{max})$  is equivalent to the OP. For  $\delta > 1$ , solving the OP does not solve  $\text{FAP2}(\delta, T_{max})$  unless the network transformation presented in Section 2.3.6 is used.

### 2.4 *Recommendations for static routing problems with fixed appointment times*

In this study of static routing problems with fixed appointment times, it was shown that offering appointment times can:

- increase the amount of time required to visit a given number of customers, and
- decrease the number of customers it is possible to visit, given a fixed duration.

The magnitude of the increase in tour duration, and the magnitude of the decrease in customers served, are both functions of the parameter  $\delta$ . As  $\delta$  increases, the amount of time required to visit a number of customers increases. Also as  $\delta$  increases, the number of customers that can be visited potentially decreases. Thus, if choosing to implement fixed appointment times, an organization should be careful with their selection of the parameter  $\delta$  in order to not over-constrain their resources. For example, suppose the resource is a truck driver, limited by an eight-hour workday, and deliveries require 40 minutes on average. Choosing  $\delta = 60$  minutes would bound the number of customer deliveries that could be performed in an 8-hour day by 8, when it may be possible to make up to 12 deliveries.

Fixed appointment times may not be appropriate when there is a high level of uncertainty in some problem parameters, such as travel times, visit times, service times, unloading times, etc. The motivation behind offering the customer fixed appointment times

is for there to be a guarantee that service will begin at the time specified. Appointments may be missed if upstream variations in process times delay the resource in its execution of subsequent tasks.

An organization that chooses to implement fixed appointment times may experience a significant increase in cost. Thus, there should be some benefit, either financial or otherwise, to justify the customer service offering. We believe that appointment times show the most benefit in applications where the customer is required to be present during the service. This occurs, for example, in home health, home grocery delivery, and home repair services. In these cases, the customer may be willing to pay a premium for the guarantee of exactly on-time service.

In the next chapter, we explore the use of fixed appointment times in home health nurse scheduling. The variation in service times of patient visits is relatively low, because there is a common set of services that home health patients typically receive. Examples include wound changes and administration of IV medications. Also, the geographic service region that a single nurse serves is typically compact enough that travel time estimates are fairly accurate as well.

## CHAPTER III

### HEURISTICS FOR DYNAMIC PERIODIC HOME HEALTH NURSE SCHEDULING PROBLEMS

#### 3.1 *Introduction*

Home health care is the business of delivering professional health care services in the home that have been ordered by a physician. The National Association for Home Care and Hospice estimates there were approximately 17,700 providers of home care nationwide in 2005, and 7.6 million patients with projected total annual expenditures of \$53.4 billion [36]. Patients who are disabled, elderly, and chronically or terminally ill are among the groups most often using home care services [17]. Demand for home care from the elderly population is expected to double by 2030 as the baby boomer generation ages [43]. Furthermore, by 2020, a 20% gap between the supply of skilled nurses and the demand for their services is expected [12]. One way to narrow this gap is to increase the productivity of existing resources. In this chapter, we study the problem of improving resource utilization by focusing on improvements to home health nurse routing and scheduling subject to patient service constraints. The optimization problems that arise in this context are dynamic variants of periodic routing and scheduling problems, where each patient visit must be assigned a precise appointment time in advance from a fixed menu of allowable times.

We define home health nurse routing and scheduling (HHNRS) problems for a set of patients that need to be visited according to a prescribed weekly frequency for a prescribed number of consecutive weeks during a planning horizon. The weekly visits for each patient must occur according to an allowable visit day combination for that patient. For example, allowable combinations for a two visit per week patient could be {Monday, Wednesday} or {Tuesday, Thursday}. Furthermore, each visit to the patient must be assigned a precise appointment time, chosen from a fixed menu of equally-spaced appointment times, such as {8:00, 8:15, 8:30,...}. In a real-world setting, it is possible that a nurse would be allowed



to enter a patient’s home early if he arrived early and the patient were ready to be seen. However, a visit would not be scheduled to begin at an irregular time like 9:23 am.

A solution to a HHNRS problem assigns patients to visit day combinations and precise appointment times on each visit day, and assigns these visits to nurses (thus generating routes) for each day of the planning horizon. Each route must start and end at a nurse’s home location and conform to shift duration constraints. The duration of a shift is comprised of visit time, travel time, and idle time. Idle time is induced when the nurse arrives to a patient location before that patient’s visit is scheduled to begin. The primary objective of HHNRS problems is to maximize nurse productivity subject to patient service constraints, measured by the number of patients served per nurse per time. Good home health nurse schedules also possess a number of desirable characteristics for patients; namely, patient visit schedules which are repeatable from week to week with respect to visit days, visit times, and service providers.

Variants of HHNRS problems that model visit day assignment decisions as exogenous decisions and consider a static set of patients, all known *a priori*, can be solved using techniques for multiple traveling salesperson problems with time windows ( $m$ -TSTPWs). However, most real-world problems are more complex *periodic* variants of the  $m$ -TSPTW. These problems model the assignment of patients to visit day combinations and nurses as endogenous decisions, and are further complicated when visit days and times must stay constant from week to week, and when the same nurse must visit the patient each time. Also, the most realistic patient setting for this periodic routing problem is dynamic: the set of patients to be served varies from week to week, and future patient arrivals are not known. Patients become known typically on the day of their requested admission visit. The scheduling decisions made for patients during any planning period affect nurse availabilities in future periods, yet they must be made without knowledge of the locations or requested service dates of future patients. With a frequently changing patient base, visit day, appointment time, and nurse assignment decisions for current patients should maintain schedule flexibility with respect to adding future patients. Unlike previous home health nurse routing and scheduling problems addressed in the literature, we study home health

routing and scheduling problems that are dynamic, periodic, and use precise appointment times for patient visits.

In this chapter, we develop a scheduling procedure that works well for a single-nurse problem variant, where a single provider serves all patient requests originating in some service region. The metropolitan home care provider that has motivated this research uses contracts with home health nurses that specify their service area boundaries, which could overlap with other nurses' areas but often do not. We also enforce strict repeatability of visit days and times throughout a patient's duration of care. While real-world problems are often a bit less rigid, we believe that developing a scheduling procedure that works well in this highly constrained environment for a single nurse represents a useful building block for more complex multiple-nurse scheduling variants.

We develop a rolling horizon myopic planning approach for the single-nurse periodic HHNRS problem with a dynamic patient set. Within the approach, we develop and implement a new capacity-based insertion heuristic that explicitly considers remaining available time in the nurse's schedules when inserting currently revealed requests in an attempt to preserve capacity for inserting future patient requests. A computational study compares the performance of the capacity-based insertion heuristic and a traditional distance-based insertion heuristic on a variety of test problems. On average, the capacity heuristic produces schedules which accept 4% more patients and perform 4% more visits per day than the distance heuristic, while requiring an average of 8.7% additional minutes of travel per visit.

In Section 3.2, we review literature related to home health nurse routing and scheduling problems. In Sections 3.3 through 3.6, we define the problem and present the solution approaches. Finally, in Sections 3.7 and 3.8, we present the results of a computational study and make concluding remarks.

### ***3.2 Literature review***

The problem we study is a dynamic periodic routing problem with application in home health nurse routing and scheduling. Therefore we review recent work in the areas of home

health scheduling, periodic routing problems, and dynamic routing problems.

### 3.2.1 Home health nurse scheduling problems

The problems most frequently addressed in the home health nurse routing and scheduling literature are often modeled as multiple traveling salesperson problems with time windows ( $m$ -TSPTWs). Visit day combination assignment decisions are assumed to be given, which implies a fixed set of patients that need to be visited each day by one more more nurses with workday length constraints. The problems are not periodic because visit day assignment decisions are not required and visit times are not required to be repeatable from week to week, and they are not dynamic because all patients are known in advance. Researchers instead focus on various side constraints involved in the patient-to-nurse assignment decisions, such as skill level requirements and patient preferences for a particular nurse.

The approach developed in Begur et al. [6] uses a simple route construction approach based on the savings and nearest neighbor heuristics to determine routes for each nurse for each day of the planning horizon. Schedulers can use a GIS map-based visual interface to modify the routes to address concerns such as balanced workload across days and matching nurse skill levels to patient needs. Eveborn et al. [21] also develop an interactive tool that creates initial  $m$ -TSPTW solutions that maximize the number of patients served and minimize distance traveled subject to hard constraints for critical time windows and soft constraints for patient provider preferences. Schedules are determined by applying a heuristic solution approach to a set partitioning model, and schedulers can make new solutions by adding staff members or relaxing constraints.

Akjiratikarl et al. [3] develop nurse routes by using particle-swarm optimization in combination with local improvement techniques to solve a tightly constrained  $m$ -TSPTW each day of the planning horizon. Time windows are formed around exogenously-specified appointment times and vary between plus or minus 5 and 15 minutes, but constraints on patient provider preferences and nurse skill level requirements are not included. Bertels and Fahle [9] use a hybrid heuristic that combines techniques such as simulated annealing, tabu search, and constraint programming to assign staff to patient visits. They strictly

enforce skill level requirements, work time limitations, and vital time windows, and model less critical time windows and various patient and provider preferences through the use of a single penalty function. Once visit-to-staff assignment decisions have been made, a hybrid linear and constraint programming model is used to optimize each staff member’s daily work plan.

The research described above does not address the complicating dynamic and periodic aspects of the home health nurse scheduling problem. One recent dissertation from Germany partially incorporates the periodic component by addressing the visit day assignment decision. Steeg [42] uses constraint programming and adaptive large neighborhood search to construct a set of schedules for each day of the week that satisfy visit day combination constraints for patients which are known in advance. The paper also partially addresses problem dynamics by developing a tabu search algorithm that can be used to incorporate periodic visits for a new patient into partial schedules. However, the repeatability of visit times throughout the patient’s duration of care is not enforced, and same day request arrivals are not handled. The problem studied does not fully address the periodic or dynamic components.

### **3.2.2 Periodic routing problems**

In the classic Period Vehicle Routing Problem (PVRP), the classic single-period Vehicle Routing Problem (VRP) is extended to multiple periods, where each customer must be visited by a vehicle a number of times over a given study period using a selection from a set of allowable visit day combinations for each customer. The problem is to simultaneously assign visit days to customers and to create daily vehicle routes for each day of the planning period that minimize total travel costs. Variants of periodic routing problems may also include time constraints, such as tour duration constraints or customer time windows.

Two early papers on periodic routing problem variants that achieved especially good computational results on test problems are Chao et al. [15] and Cordeau et al. [19]. The first paper, studying the periodic traveling salesman problem (PTSP), uses a binary integer program to assign customers to visit days and solves the resultant TSP on each day of the

planning horizon using a modified Clarke-Wright algorithm. Local search methods are used to improve the initial visit day assignments and routes. Alternatively, Cordeau et al. [19] presents a tabu search algorithm for the PVRP that relies on very few parameters and uses a generalized-insertion heuristic to perform route construction and improvement.

More recent examples of work on the PTSP include Bertazzi et al. [8] and Polacek et al. [38]. In the first paper, a cheapest-insertion method is used to assign visit day combinations to customers and insert them into corresponding tours. Routes are improved through a combination of local search techniques, such as removing a set of scheduled customers from their tours, assigning them to new visit day combinations, and inserting them into corresponding tours. In the second paper, a variable neighborhood search heuristic is used to assign visit days to customers and determine tours. Penalty costs are used to enforce consistency of inter-visit time for individual customers over the planning horizon. Hemmelmayr et al. [26] present a variable neighborhood search algorithm that obtains competitive results for both the PVRP and PTSP; the proposed heuristic often outperforms other methods.

Other recent work extends the periodic routing literature to allow customer service frequency to be a decision variable. Francis et al. [23] study the PVRP with service choice, where all customers have a minimum frequency with which they require visits but will accept more frequent visits. Baptista et al. [4] study a waste collection problem where feasible visit frequencies and patterns are defined by customer demand rates. Both papers develop heuristic methods to assign customers to visit frequencies and visit day combinations before developing routes for each day of the planning horizon.

In research surveyed above, methods are developed to create routes that visit each customer on regular visit days. However, no research to date investigates problems that constrain the visit times on those days to be consistent or that customers are visited by the same vehicle from week to week; these features are important in home health nurse scheduling problems. The periodic routing literature also assumes all customer requests for service are known in advance.

### 3.2.3 Dynamic routing problems

Customer demands are revealed over time in dynamic extensions to routing problems. In stochastic variants of dynamic routing problems, it is assumed that probabilistic information describing future customer requests is known when planning. Existing research has focused on solution heuristics that perform route planning throughout the planning horizon to develop routes that serve known customers and are flexible to serving future arriving customers.

In some cases, the planning process is event-driven, where decisions are made each time an event (*e.g.*, new customer arrival) occurs. The approach in Ichoua et al. [32] uses a tabu search heuristic to plan vehicle routes that serve a set of customers which have time windows. Each route specifies a planned sequence of customers, and a vehicle is committed to a specific customer only after it has departed for that customer. The vehicle will not leave its current location if it will arrive at the next customer location early. This waiting strategy implicitly accounts for future request arrivals by preserving the ability to incorporate new request arrivals. A separate heuristic is used to determine when a vehicle should depart for the next customer, based on the probability of a new customer arriving. Similarly, van Hemert and LaPoutre [44] develop an evolutionary approach that is used to determine whether a vehicle should wait in a region that is likely to produce a new customer. Their approach allows anticipatory moves to new regions. Bent and van Hentenryck [7] propose a multiple plan approach for dynamic routing problems that maintains a set of alternate routing plans at each execution step for the currently arrived customer requests. When a new customer request arrives, the approach selects a plan that can feasibly incorporate the new customer. For a dynamic and stochastic routing problem variant, they extend the multiple plan approach to a multiple scenario approach, where each scenario contains simulated future requests in addition to the currently arrived customer requests.

An alternative to event-driven planning approaches is to divide the planning horizon into discrete time intervals at which to perform planning, such as in Hvattum et al. [31]. This paper formulates a multistage stochastic model, develops a hedging solution method to generate sample scenarios, and solves each heuristically as a static VRP. Common features

from the solutions to each scenario are combined to form a solution to the original problem. At each planning interval, the expected cost of serving all customers is minimized. The paper shows there is some advantage to using planning intervals wide enough to allow batching of customers at each stage.

In each of the above papers, final decisions regarding customer service requests are delayed as much as possible. Both the customers assigned to a vehicle and the sequence of those customers can change at any time, as long as the service windows are not violated for any associated customers. In the problem we study, the set of decisions that are fixed at each execution step is larger. Appointment time decisions for the entire duration of a customer's service horizon are made upon arrival, and those decisions cannot change. It is not clear whether the methods surveyed above would be as effective in such a highly constrained environment.

### 3.3 *Problem statement*

We now define the dynamic periodic single-home-health-nurse routing and scheduling problem. For the purposes of this chapter, we define the problem for a finite study period of  $P$  days. Although we recognize that the real-world problem continues beyond this period, we will attempt to minimize horizon end effects in our computational study.

**Nurse:** There is a single nurse, only available certain days of the week  $\mathcal{K} \subseteq \{1, 2, \dots, 7\}$ , that can perform patient visits in a defined geographic service region. The nurse has a home location  $n_0$  where his daily routes will each begin and end. Each working day, the nurse has a fixed menu of equally-spaced appointment times available for scheduling patient visits,  $\mathcal{M} = \{a_d + j\delta : j = 0, 1, \dots, m_d\}$ , where  $a_d$  is the earliest appointment time,  $m_d + 1$  is the total number of appointments available on day  $d$ , and  $\delta$  is the spacing between appointment times. For a typical setting found in practice that is the basis for our test problems,  $a_0 = 8:00$ ,  $\delta = 15$  minutes, and  $m = 34$ , such that the set of scheduled visits times that are possible are  $\mathcal{M} = \{8:00, 8:30, \dots, 4:30\}$ . Note that this appointment time menu implicitly limits the maximum shift duration for the nurse, measured in hours from the time the first patient visit begins until the time the last patient visit ends.

**Patients:** There is a set of patient locations  $\mathcal{N} = \{1, 2, \dots, n\}$  from which requests, which are not known in advance, can arise. Travel is allowed between all pairs of patient locations and the nurse home location, and is represented by  $\mathcal{A} = \{(i, j): i, j \in \mathcal{N} \cup n_0\}$ , with the travel time in hours on  $(i, j)$  denoted by  $h(i, j)$ . Each patient has an arrival date and time  $a_i = (d_i, t_i)$  when the request for service becomes known. Each patient has an associated integer service frequency  $f_i$  measured in visits per week. Each also has an episode of care duration  $\Delta_i$ , which is the number of consecutive weeks forward that she will receive regular weekly visits. Each visit to patient  $i$  has a duration  $w_i$  in hours, and each patient has a set of allowable visit day combinations  $\mathcal{V}_i$  on which she can be visited. For example, patient  $i$  with  $f_i = 2$  might have  $\mathcal{V}_i = \{(1, 3), (2, 4), (3, 5)\}$ .

**Patient visits:** The total number of visits a patient requires is  $1 + L_i$ , where  $L_i = f_i \Delta_i$ . Each visit to a patient must be scheduled an appointment time from the menu  $\mathcal{M}$ . We will use  $l_i^j = (d_i^j, t_i^j)$  to denote visit times, where  $j = 0$  corresponds to the admission visit and  $j = 1, \dots, L_i$  correspond to the regular repeatable weekly visits. The patient's admission visit must take place within 24 hours of time the request becomes known. The regular visits must take place on the same days throughout the episodes of care. Regular visits occurring on the same weekday must occur at the same appointment time from week to week, but visits for a given patient that occur on different days of the week are not required to occur at the same time. For example, patient  $i$  that has been assigned visit day combination  $\{1, 3\}$  may be visited at 8:00 a.m. each Monday and 10:00 a.m. each Wednesday for  $\Delta_i$  consecutive weeks.

**Routes:** Each route  $\sigma_d$  for day  $d$  in the horizon begins at the nurse's home, visits each patient receiving service on day  $d$  at her scheduled appointment time, and ends at the nurse's home.

**Planning intervals:** The study period is divided into planning intervals  $\mathcal{G}$ . Each  $g \in \mathcal{G}$  is a vector  $(d_g, t_g)$  specifying the date and time of the end of the planning interval. Denote the set of new known patient requests received for any planning interval  $g$  as  $\mathcal{P}_g$ , defined in Equation (33), where  $time(d, t)$  is a function that returns the time in minutes from some standard reference point:



$$\mathcal{P}_g = \{i \in \mathcal{N} : \text{time}(d_{g-1}, t_{g-1}) < \text{time}(d_i, t_i) \leq \text{time}(d_g, t_g)\}. \quad (33)$$

**Solutions:** The problem is to assign patients to visit day combinations and precise appointment times on each visit day, thus establishing routes  $\sigma_d$  for each day  $d$  of the study period,  $d = 1, 2, \dots, P$ . Patients which are not successfully scheduled are rejected and denied service from this nurse. Each induced route  $\sigma_d$  is feasible only if the travel times between consecutive patient visits are short enough so that the nurse may arrive in time for the next visit.

**Objective:** The objective is to schedule as many patient visits as possible during the study period. This is not equivalent to maximizing the number of patients served, as patients require different numbers of visits. Note that a related objective of minimizing total nurse travel and idle time between patients will maximize the time that the nurse can spend visiting patients.

### 3.4 *Rolling horizon myopic planning approach*

We develop a rolling horizon planning framework for the problems defined in Section 4.3, which are dynamic periodic problems with fixed appointment times selected from a menu of equally-spaced times. We also develop two heuristics to use within this framework to make admission and scheduling decisions for the set of new patients at each planning interval. The first is a distance heuristic that uses a traditional distance-based insertion cost that has been adapted for the periodic problem. It selects visit insertion options that minimize the total travel distance induced by all visits required for the patient being scheduled. The second is a capacity heuristic that minimizes both the travel distance induced between pairs of patient locations, and the idle time induced by arriving to a patient location earlier than the scheduled appointment time. This more complex heuristic attempts to preserve capacity for inserting future patients, selecting appointment times for current patients by explicitly considering the remaining time between all consecutive pairs of patient appointments in the nurse's current schedule. The heuristics are described in Sections 3.5 and 3.6.

For this study, we assume that there is one planning interval per day that occurs at

the beginning of the workday, during which routes  $\sigma_d$  are planned for the current day and partially planned for some number of subsequent days. We assume that the visit duration and episode of care is the same for each patient,  $w_i = w$  and  $\Delta_i = \Delta \forall i \in \mathcal{N}$ . Additionally, we assume that  $w$  is a multiple of the grid spacing parameter  $\delta$ . We make these assumptions primarily for simplicity of exposition, because the ideas can be easily extended to cases where multiple planning intervals occur each day and visit durations and episodes of care are patient-dependent.

The rolling horizon myopic planning approach makes admission decisions regarding new known patients each planning interval and updates a partially planned nurse route for each day of the **planning horizon**. The planning horizon is the number of days for which we plan routes  $\sigma_d$  at each planning interval. Each newly admitted patient  $i$  needs regular visit appointments for the next  $\Delta$  weeks, and these visits must be incorporated into partial nurse schedules that serve previously admitted patients. If a new patient is admitted in the current interval, she will remain in the schedule until the end of her episode of care  $\Delta + 1$  weeks later; she receives an admission visit during the first week, and then regular visits during the following  $\Delta$  weeks. Thus, we can use a shortened planning horizon of  $T$  days, sufficient to establish the entire appointment schedule for any patient until the end of their episode of care, where  $T = |\mathcal{K}|(\Delta + 1)$ . We will plan routes  $\sigma_u$  for days  $u = d_g, \dots, d_g + T$  and denote the set of daily routes as master schedule  $\mathcal{S}_g = \{\sigma_u : u \in \{d_g, d_g + 1, \dots, d_g + T\}\}$ .

As the rolling horizon procedure transitions from planning interval  $g$  to  $g + 1$ , the route specified by  $\sigma_{d_g}$  is executed and is no longer needed in the master schedule for the new period,  $\mathcal{S}_{g+1}$ . The master schedule for the new period is created as described in Equation (34):

$$\mathcal{S}_{g+1} = \mathcal{S}_g \setminus \sigma_{d_g} \cup \sigma_{d_g+T+1}. \quad (34)$$

The current routes for operating periods  $d_g + 1$  through  $d_g + T$  are moved into the new initial master schedule and an empty route for day  $d_g + T + 1$  is added. The planning process then repeats as the newly arrived patients in  $\mathcal{P}_{g+1}$  are added to the routes in  $\mathcal{S}_{g+1}$ .

Figure 7 illustrates what a master schedule may contain when planning is performed

		Appointment times $q_j, j=1,\dots,12$											
		1	2	3	4	5	6	7	8	9	10	11	12
Week	0	B			A		C						E
	1	B			A		C				E		
	2	B									E		
	3	B									E		
	4										E		

**Figure 7:** Snapshot of master schedule at planning interval  $g$

in a given planning interval. It depicts a snapshot of patient appointments for a single weekday, say Monday, across  $\Delta + 1$  weeks, beginning in the week of the current planning interval, taken here to be week 0. Each row represents Monday's visit schedule for the corresponding week number. Each column represents an appointment time available from the appointment time menu, e.g.  $q_1 = 8:00$  a.m. Appointment times to which patients have been previously assigned are shaded in dark gray. In this example, each patients' episode of care is 4 weeks. Patient  $B$  was admitted in week -1, and will continue receiving regular visits at time  $q_1$  through week 3. Patient  $A$  was admitted in week -3 and will continue receiving regular visits at time  $q_4$  through week 1. Patient  $E$  is processed during the current planning interval. The patient is assigned an admission visit at time  $q_{12}$  in week 0 and regular visits at time  $q_{10}$  in weeks 1 through 4.

The framework of the rolling horizon myopic planning approach is outlined in Algorithm 4. Section 3.4.1 gives a detailed explanation of the subroutine used to make scheduling decisions regarding new patient arrivals in each planning period.

---

**Algorithm 4** Rolling horizon myopic planning approach

---

- 1: **for all**  $g \in \mathcal{G}$  {at the beginning of each planning interval} **do**
  - 2:   create  $\mathcal{S}_g$  from  $\mathcal{S}_{g-1}$
  - 3:   use Algorithm 5 to make admission and scheduling decisions for the patients in  $P_g$
  - 4:   insert visits for accepted new patients into  $\mathcal{S}_g$
  - 5:   execute  $\sigma_{d_g}$
  - 6: **end for**
-

### 3.4.1 Route updating procedure

Algorithm 5 is used to update routes at each planning epoch given new patient arrivals. The procedure schedules appointments for as many of the new patients in the given planning interval as possible in order of cheapest total insertion cost. Here, we use a general cost function  $c(i, q)$  to represent the cost of assigning appointment time  $q$  for a single visit to patient  $i$ .  $C_i$  is the sum of  $c(i, q)$  across all visits to patient  $i$ . In this greedy approach, as long as a feasible set of appointment times can be identified for at least one patient in  $\mathcal{P}_g$ , the patient with the lowest total insertion cost will be admitted and inserted into the master schedule  $\mathcal{S}_g$ .

---

**Algorithm 5** Route updating procedure

---

```

1: Initialize:  $C_i = \infty \forall i \in \mathcal{P}_g$ 
2: if  $\mathcal{P}_g \neq \emptyset$  then
3:   for  $i \in \mathcal{P}_g$  do
4:     find cheapest feasible admission appointment and appointment time combination
       for patient  $i$ 
5:   end for
6:   let  $C^* = \min_i C_i$ ;  $i^* = \arg \min_i C_i$ 
7:   if  $C^* < \infty$  then
8:     insert  $i^*$  into cheapest feasible admission appointment and appointment time com-
       bination
9:      $\mathcal{P}_g = \mathcal{P}_g \setminus \{i\}$ 
10:    return to 2.
11:   else
12:     STOP
13:   end if
14: else
15:   STOP
16: end if

```

---

Sections 3.4.2 - 3.4.5 discuss how feasible admission appointment times and appointment time combinations are identified, and how their cost is evaluated.

### 3.4.2 Appointment time feasibility

A nurse's workday can be spent driving, waiting, or visiting patients in up to  $m + 1$  appointment time slots. Let  $\mathcal{Q}$  denote all potential appointment times in the nurse's daily schedules, beginning in the day of the current planning interval  $d_g$  and extending through

**Table 2:** Predecessor and successor times and locations of  $q_8$  in Figure 7

Week	$p_{q_8}$	$o(p_{q_8})$	$s_{q_8}$	$o(s_{q_8})$
0	6	C	12	E
1	6	C	10	E
2	6	C	10	E
3	1	B	10	E
4	-	$n_0$	10	E

the last day of the planning horizon  $d_g + T$ . Each  $q \in \mathcal{Q}$  is associated with some date and time  $(d_q, m_q)$ , such that  $d_q \leq d_g \leq d_g + T$  and  $m_q \in \mathcal{M}$ . Let  $o(q)$  be a function that takes value  $j$  if appointment time  $m_q$  on date  $d_q$  has been assigned to patient  $j$  and zero otherwise. Recall that a nurse's schedule can not be rearranged to accommodate a new patient because existing visits for previously admitted patients are fixed. Therefore, if we wish to consider assigning appointment time  $q$  to some visit for patient  $j$ , we must ensure it will be feasible with respect to the direct predecessor and successor of  $q$  in route  $\sigma_{d_q}$ .

The predecessor appointment time of  $q$  on route  $\sigma_{d_q}$  is the latest appointment time prior to  $q$  having a patient assigned to it. The successor appointment time is the earliest appointment time after  $q$  having a patient assigned to it. In either case, if no such patient appointment exists,  $q$  then indicates that the previous (or next) location is the home location. Denote the predecessor appointment time and patient location as  $p_q$  and  $o(p_q)$ , and the successor appointment time and patient location as  $s_q$  and  $o(s_q)$ . Referring back to Figure 7 for an example, Table 2 gives the predecessor and successor times and locations of  $q_8$  in weeks 0 through 4.

Appointment time  $q \in \mathcal{Q}$  is a **feasible appointment time** for patient  $i$  if and only if it is feasible to serve the predecessor patient, travel to  $i$ , serve patient  $i$ , and travel to the successor patient, without arriving late to any of those patient visits. If the predecessor location is the nurse's home (first visit of the day), then it is assumed to always be feasible to arrive on time at patient  $i$ . Furthermore, if the successor location is the nurse's home (last visit of the day), it is again assumed to be feasible to arrive home at any time. The set of appointment times meeting these conditions for patient  $i$  is denoted as  $\mathcal{Q}_i$  and defined formally by Equation (35):

$$\begin{aligned}
\mathcal{Q}_i &= \{q : q \in \mathcal{Q}\} \cap \{q : o(q) = 0\} \cap \\
&(\{q : o(p_q) = n_0\} \cup \{q : \text{time}(d_{p_q}, t_{p_q}) + w + h(o(p_q), i) \leq \text{time}(d_q, m_q)\}) \cap \\
&(\{q : o(s_q) = n_0\} \cup \{q : \text{time}(d_q, m_q) + w + h(i, o(s_q)) \leq \text{time}(d_{s_q}, t_{s_q})\}).
\end{aligned} \tag{35}$$

### 3.4.3 Admission appointment time feasibility and cost

Appointment time  $q \in \mathcal{Q}_i$  is a **feasible admission appointment time** for patient  $i$  if and only if it begins within 24 hours of the planning interval  $g$  in which patient  $i$  is processed. A cheapest insertion approach is used to select among alternative feasible admission appointment times for a patient. The set of appointment times meeting these conditions is denoted as  $\mathcal{Q}_i^a$ , and the selected admission appointment time is denoted as  $q_0^*$ . Equations (36) and (37) describe how the set of feasible admission appointments is constructed and how the cheapest admission appointment time is selected:

$$\mathcal{Q}_i^a = \{q \in \mathcal{Q}_i : \text{time}(d_g, t_g) \leq \text{time}(d_q, m_q) \leq \text{time}(d_g + 1, t_g)\}, \tag{36}$$

$$q_0^* = \arg \min_{q \in \mathcal{Q}_i^a} c(i, q). \tag{37}$$

### 3.4.4 Regular appointment time feasibility and cost

Appointment time  $q \in \mathcal{Q}_i$  is a **feasible regular appointment time** for patient  $i$  if and only if it occurs in the first full week following the current planning interval, is repeatable on the same days and times each week throughout patient  $i$ 's episode of care, and it appears in some allowable visit day combination for patient  $i$ . Let  $\mathcal{D}_i$  contain all days of the week specified by the allowable visit day combinations in  $\mathcal{V}_i$ . Let  $\text{week}(d, t)$  be a function that returns the weeks elapsed from some standard reference point given a date and time. The set of appointment times meeting regular visit feasibility conditions for patient  $i$  is denoted as  $\mathcal{Q}_i^r$  and is described by Equation (38):

$$\begin{aligned} \mathcal{Q}_i^r = & \{q : (d_q + y|\mathcal{K}|, m_q) \in \mathcal{Q}_i, \forall y = 0, \dots, \Delta - 1\} \cap \\ & \{q : d_q \in \mathcal{D}_i\} \cap \{q : \text{week}(d_q, t_q) = \text{week}(d_g, t_g) + 1\}. \end{aligned} \quad (38)$$

A feasible regular appointment time consists of a set of  $\Delta$  appointment times that are feasible for patient  $i$  in consecutive weeks. The day of week and time of day is the same for each of the  $\Delta$  appointment times comprising the regular appointment time. As seen in Figure 7 and Table 2, the predecessor and successor appointment times and locations may be different from week to week. The following two observations establish that during execution of the new patient processing procedure for period  $g$ , the search for feasible regular appointment times can be restricted to those appointment times which are feasible in  $\text{week}(d_g, t_g) + 1$ .

**Observation 1.** *During planning interval  $g$ , if  $q = (d_q, m_q) \in \mathcal{Q}_i$  and  $\text{week}(d_q, m_q) = \text{week}(d_g, m_g) + 1$ , then  $(d_q + y|\mathcal{K}|, m_q) \in \mathcal{Q}_i$  for  $y = 1, \dots, \Delta - 1$ .*

Observation 1 follows from the manner in which visit schedules are constructed during the rolling horizon planning procedure. Let  $z = \text{week}(d_g, t_g)$ . During planning period  $g$ , the only patients which can present schedule conflicts in weeks  $z + 1$  through  $z + \Delta$  are those patients which were admitted during weeks  $z - \Delta + 1, \dots, z$ . If some previously admitted patient presented a scheduling conflict in week  $z + y$  for  $1 < y \leq \Delta$ , but not in week  $z + 1$ , the visit to that patient in week  $y$  would not meet the repeatability condition necessary for regular appointment times to be feasible.

**Observation 2.** *During planning interval  $g$ , if  $q \notin \mathcal{Q}_i$  for some  $q$  where  $\text{week}(d_q, t_q) = \text{week}(d_g, t_g) + 1$ , then  $q \notin \mathcal{Q}_i^r$ .*

Observation 2 follows directly from the regular visit feasibility conditions specified in Equation (38).

According to Observation 1, if some appointment time  $q$  is feasible for a visit to patient  $i$  in its first regular visit week, the appointments on those same days and times will be feasible for patient  $i$  in each of her regular visit weeks. This is independent of whether

the predecessor or successor appointment times and locations vary in subsequent weeks. Furthermore, according to Observation 2, restricting the search to the first regular visit week does not exclude any feasible regular visit times.

The cost of one regular appointment time is defined to be a  $\Delta$ -week sum of appointment time costs that is dependent on the predecessor and successor appointment times and locations in each week. The cost of one feasible regular appointment time is denoted as  $c_r(i, q)$  and is calculated as specified in Equation (39):

$$c_r(i, q) = \sum_{y=0}^{\Delta-1} c(i, x) \text{ where } x = (d_q + y|\mathcal{K}|, m_q). \quad (39)$$

### 3.4.5 Appointment time combination feasibility and cost

A set of  $f_i$  feasible regular appointment times forms a **feasible appointment time combination** (denoted as  $O$ ) for patient  $i$  if and only if each appointment time occurs on a unique day of some allowable visit day combination  $v \in \mathcal{V}_i$ . The set of feasible appointment time combinations for patient  $i$  is denoted as  $\mathcal{O}_i$  and is described by Equation (40), where  $dOW(q)$  returns the day of week of appointment time  $q$ :

$$\mathcal{O}_i = \{O = (q_1, \dots, q_{f_i}) : (dOW(q_1), \dots, dOW(q_{f_i})) \in \mathcal{V}_i; q_j \in \mathcal{Q}_i^r \ \forall j = 1, \dots, f_i\}. \quad (40)$$

The cost of a feasible appointment time combination  $O$  is the sum of the costs of the  $f_i$  regular appointment times included in the combination. This cost is denoted as  $c(O)$ , and is calculated as specified in Equation (41), and the cheapest feasible appointment time combination is selected as specified in Equation (42):

$$c(O) = \sum_{q_k \in O} c_r(i, q_k), \quad (41)$$

$$O^* = \arg \min_{O \in \mathcal{O}_i} c(O). \quad (42)$$

Note that selecting the cheapest appointment time combination  $O^*$  implies a selection of  $f_i$  feasible regular appointment times.



Recall that  $l_i^j$  for  $j = 0, \dots, L_i$  denotes the selected appointment times for patient  $i$ . The admission appointment time  $l_i^0$  corresponds to the cheapest feasible admission appointment,  $q_0^*$ . The regular appointment times for patient  $i$  are obtained by arranging the regular appointment times in  $O^*$  in the order they occur during the week. The appointment times assigned to patient  $i$  are specified by Equations (43) - (44), and the total cost of inserting patient  $i$  is given by Equation (45):

$$l_i^0 = q_0^*, \quad (43)$$

$$l_i^{j+yf_i} = (d_{q_j} + y|\mathcal{K}|, t_{q_j}), \quad \forall q_j \in O^*, \quad y = 0, \dots, \Delta - 1, \quad (44)$$

$$C(i) = \sum_{j=0}^{L_i} c(i, l_i^j) = c(i, q_0^*) + c(O^*). \quad (45)$$

Sections 3.5 and 3.6 define specific cost functions  $c^{DH}$  and  $c^{CH}$  which replace  $c$  in the distance and capacity heuristics, respectively. Note that both heuristics use the rolling horizon framework, and both use greedy cheapest insertion. The heuristics differ in their determination of “cheapest.”

### 3.5 Distance heuristic

The distance heuristic (DH) uses a traditional distance-based insertion cost to determine patient scheduling decisions, where distance here is equivalent to the travel time  $h$  between locations. The cost of inserting a single visit to patient  $i$  at appointment time  $q$  is based on the distance to the predecessor and successor patient locations. The cost of selecting appointment time  $q$  as a regular visit is a sum of those single visit costs over  $\Delta$  weeks, where the predecessor or successor patient location may change from week to week as patients reach the end of their episodes of care. The single visit cost and multiple week cost are defined in Equations (46) and (47):

$$c^{DH}(i, q) = h(o(p_q), i) + h(i, o(s_q)) - h(o(p_q), o(s_q)), \quad (46)$$

$$c_r^{DH}(i, q) = \sum_{y=0}^{\Delta-1} c^{DH}(i, x) \text{ where } x = (d_q + y|\mathcal{K}|, m_q). \quad (47)$$

Following Algorithm 5, in each planning interval  $\mathcal{P}_g$ , the distance heuristic identifies the cheapest admission appointment time and the cheapest regular appointment time combination for all arriving patient requests. Then, it inserts visits for the patients with cheapest total insertion cost, until no additional patient visits can be scheduled.

Algorithm 6 specifies the method that the distance heuristic uses to select the cheapest admission appointment and regular appointment combination for each patient.

---

**Algorithm 6** Cheapest admission appointment and regular appointment combination for patient  $i$

---

- 1:  $c^* = \min\{c^{DH}(i, q) : q \in \mathcal{Q}_i^a\}$
  - 2:  $\Phi = \{q \in \mathcal{Q}_i^a : c^{DH}(i, q) = c^*\}$
  - 3:  $q_0^* = distanceTieBreak(\Phi)$
  - 4: **for**  $k \in \mathcal{D}_i$  **do**
  - 5:    $c^*(k) = \min\{c_r^{DH}(i, q) : (q \in \mathcal{Q}_i^r) \text{ AND } (dOW(q) = k)\}$
  - 6:    $\Phi = \{q \in \mathcal{Q}_i^r : (dOW(q) = k) \text{ AND } (c_r^{DH}(i, q) = c^*(k))\}$
  - 7:    $q^*(k) = distanceTieBreak(\Phi)$
  - 8: **end for**
  - 9:  $v^* = \arg \min_{v \in \mathcal{V}_i} \{\sum_{q^*(k) \in v} c^*(k)\}$
- 

Algorithm 6 identifies the cheapest regular appointment time  $q^*(k)$  for each day of the week and the best visit day combination  $v^*$  formed by selecting a subset of those cheapest regular appointment times. The cheapest regular appointment combination,  $O^*$ , is defined by  $q^*(k)$  and  $v^*$ .

Notice that the distance heuristic outlined in Algorithm 6 is hierarchical, calling a tie-breaking method specified in Algorithm 7 anytime a selection must be made between alternative appointment times which have the same cost. Consider the example in Figure 8, which illustrates why the tie-breaking method is necessary. Suppose we wish to insert a regular appointment which must repeat across  $\Delta = 4$  weeks for patient  $i$  on the day shown. By Observations 1 and 2, we can restrict the search for feasible regular appointments to those times which are feasible in week 1. Suppose the travel time between any two patient locations is greater than 0, but less than the spacing between two consecutive appointments  $q_j$  and  $q_{j+1}$ . Then, the times which are feasible for  $i$  are  $q_7$  and  $q_8$ . The insertion costs  $q_7$  and  $q_8$  are equal because the predecessor and successor pairs surrounding them are the same in each week.

		Appointment times $q_j, j=1,\dots,14$													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Week	0	A		B		C					D		E		F
	1	A		B									E		F
	2	A													F
	3														

**Figure 8:** Master schedule illustrating need for distance tie breaker

Algorithm 7 breaks ties between alternative optimal appointment times using information from the first week of scheduled appointments. First, the tie-breaking rule selects the alternative optimal appointment time that occurs earliest in the day. This is especially helpful when selecting admission appointments, because appointment times occurring later in the day remain available for patients which may arrive in future planning periods and require admission appointments on the same day. Second, if there are then additional alternative optimal appointment times that share the same predecessor and successor locations with this earliest time, an additional tie-breaker is employed; the method selects either the earliest or the latest appointment time in the interval. The earliest appointment time is selected if the patient to be inserted is nearer to the predecessor location, and the latest is selected if the patient is nearer to the successor location. In the example shown in Figure 8, the distance heuristic would select  $q_7$  if  $i$  is closer to  $C$ , and  $q_8$  if  $i$  is closer to  $D$ .

This tie breaking method may work better than a random selection among the alternative optimal appointments within the interval. Note that intervals containing multiple consecutive available time slots may accommodate multiple appointment insertions. Enforcing the adjacency between the appointment being inserted and its nearest neighbor in the current interval preserves the largest number of *consecutive* available appointment time slots in the partial schedule for inserting additional appointments.

In some cases, selecting between alternative appointment times based on proximity to the predecessor or successor location can reduce the number of available appointment times for potential future patients. To see this, consider the example shown in Figure 9, where the

---

**Algorithm 7** *distanceTieBreak*( $\Phi$ )

---

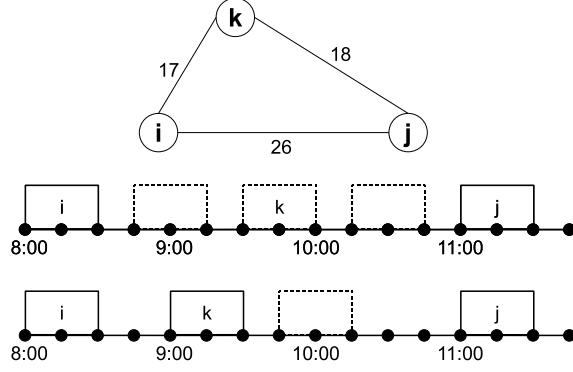
```
1: let  $e = \arg \min_{q \in \Phi} \text{time}(d_q, m_q)$ 
2:  $\Phi' = \{q \in \Phi : (p_q = p_e) \text{ AND } (s_q = s_e)\}$ 
3: if  $h(o(p_e), i) \leq h(i, o(s_e))$  then
4:    $q^* = \arg \min_{q \in \Phi'} \text{time}(d_q, m_q)$ 
5: else
6:    $q^* = \arg \max_{q \in \Phi'} \text{time}(d_q, m_q)$ 
7: end if
8: return  $q^*$ 
```

---

service time at each patient is 30 minutes and the grid spacing parameter is  $\delta = 15$  minutes. Patient  $i$  has been assigned appointment time 8:00, patient  $j$  has been assigned appointment time 11:00, and we wish to schedule a visit to patient  $k$  in one of the appointment times available between  $i$  and  $j$ . The dotted rectangles in the first schedule depict a best-case use of the capacity available between  $i$  and  $j$ . If a sequence of three patients were to arrive, such that no inter-visit travel distance would be greater than  $\delta = 15$ , those patients could be assigned appointment times 8:45, 9:30, and 10:15. Of the three appointment times which maximally utilize the capacity between  $i$  and  $j$ , the only one feasible for  $k$  is 9:30. However, the distance-tie breaking method would consider only 9:00 and 10:00, the earliest and latest feasible appointment times, and would select 9:00 because the distance from  $k$  to  $i$  is less than the distance from  $k$  to  $j$ . Once patient  $k$  is assigned to appointment time 9:00, only one additional appointment may be scheduled between  $k$  and  $j$ . Thus, the distance heuristic makes a decision in this case which reduces the total “capacity” of the time interval between appointments  $i$  and  $j$  from 3 to 2.

### 3.6 Capacity heuristic

In our dynamic periodic fixed-appointment time problem, some appointment times may be preferred for a given patient visit insertion because they preserve more flexibility for inserting appointments for future arriving patients into the set of daily schedules  $\mathcal{S}_g$ . However, such appointment times cannot necessarily be determined from the distance-based insertion costs alone, which only consider information regarding travel distance induced on the schedule and ignore idle time. We therefore develop a new insertion criterion that explicitly accounts for the impact of the fixed menu of appointment times by considering both the



**Figure 9:** Selecting  $q$  based on proximity to predecessor or successor patient location can reduce schedule capacity

travel and idle time induced between pairs of patient locations when scheduling visits for new patients.

We refer to this as a capacity-based insertion criterion, because with each decision, we attempt to preserve the capacity for inserting future appointments into the schedule by explicitly considering the remaining available appointments in the nurse’s daily schedule. For each appointment  $q \in \mathcal{Q}_i$ , we first calculate a best-case maximum number of patient visits that can be scheduled in the interval between its predecessor and successor, independent from the patient being scheduled. Then, we determine how the capacity available in the interval is affected if patient  $i$  is assigned appointment time  $q$ . The criterion selects those appointments which minimally impact the best-case capacity available prior to the insertion. This is a somewhat optimistic approach, as it uses best-case analysis to preserve as much future capacity as possible.

The capacity heuristic (CH) we develop uses a hierarchical objective function to identify minimum cost insertions on a given day as those appointment times which first minimize reduced visit capacity and reduced flexible capacity, defined in Sections 3.6.1 and 3.6.2, respectively. If alternative optimal appointment times remain, the heuristic appends the hierarchical objective function used in the distance heuristic. An additional level of hierarchy is used to balance nurse workload throughout the week when selecting regular appointment time combinations, described in Section 3.6.3.

### 3.6.1 Reduced visit capacity

The first level of the hierarchy selects appointment times which minimize the **reduced best-case visit capacity** of the set of daily schedules, defined as the number of patient visit insertions by which the best-case capacity of the total schedule is reduced if the appointment in question is selected. To calculate the reduced best-case visit capacity associated with assigning appointment time  $q$  to patient  $i$ , the best-case visit capacity of the interval between  $p_q$  and  $s_q$  must first be calculated. The **best-case visit capacity** between  $p_q$  and  $s_q$  is the maximum number of patient appointments that can be scheduled if each induced travel segment requires at most one appointment time spacing period,  $\delta$ . In the problem setting we study, appointments are spaced by  $\delta = 15$  minutes, so each travel segment in the best-case scenario is less than 15 minutes in length.

Inserting  $k \in \mathbb{Z}^+$  consecutive appointments into a partial schedule requires a minimum of  $1 + k \left( \frac{w}{\delta} + 1 \right)$  consecutive available appointment time slots. Note also that for a given interval specified by appointment  $q$ , the minimum number of appointment time slots required to travel from  $o(p_q)$  to  $o(s_q)$  through a number of intermediate patient locations is  $\left\lceil \frac{h(o(p_q), o(s_q))}{\delta} \right\rceil \delta$ . Thus, to insert  $k$  consecutive appointments in the interval between  $p_q$  and  $s_q$ , Equations (48) and (49) must be satisfied. We denote the best-case visit capacity of the interval between  $p_q$  and  $s_q$  as  $VC(p_q, s_q)$  and set its value equal to the maximum integer  $k$  satisfying Equations (48) and (49):

$$s_q \geq p_q + \delta + k(w + \delta), \quad (48)$$

$$s_q \geq p_q + \left\lceil \frac{h(o(p_q), o(s_q))}{\delta} \right\rceil \delta + kw. \quad (49)$$

Inserting some patient  $i$  into appointment time  $q$  will induce two new intervals in the nurse's schedule:  $(p_q, q)$ , and  $(q, s_q)$ . The reduced visit capacity  $RVC(i, q)$  associated with inserting patient  $i$  into appointment time  $q$  is calculated by subtracting the best-case visit capacities of the two new intervals from the visit capacity of the original interval, and adding one for patient  $i$  being inserted. This calculation is formalized in Equation (50):

$$RVC(i, q, \phi) = VC(p_q, s_q) - VC(p_q, q) - VC(q, s_q) + 1. \quad (50)$$

Of the alternative appointment times meeting the first-level criteria, the hierarchical objective function used in the capacity heuristic next prioritizes those appointment times which minimize reduced flexible capacity.

### 3.6.2 Reduced flexible capacity

Some intervals possess **flexible capacity**, defined to mean that the amount of time between  $p_q$  and  $s_q$  is strictly more than what is required to insert  $VC(p_q, s_q)$  patients, but strictly less than the amount of time needed to insert  $VC(p_q, s_q) + 1$  patients. Flexible capacity is useful when patients arrive that require more than  $\delta$  minutes of travel on some travel segment. The amount of flexible capacity between  $p_q$  and  $s_q$  is measured by the number of intermediate appointment times that will not be dedicated to visiting patients and traveling between appointments in the best-case scenario. The flexible capacity score  $FC(p_q, s_q)$  is formally defined using Equation (51):

$$FC(p_q, s_q) = \frac{s_q - p_q}{\delta} - VC(p_q, s_q) \left( \frac{w}{\delta} + 1 \right) - 1. \quad (51)$$

A unit of flexible capacity is consumed when inserting patient  $i$  at appointment time  $q$  causes the patient to be directly adjacent to either  $o(p_q)$  or  $o(s_q)$ , and the resultant travel segment  $h(o(p_q), i)$  or  $h(i, o(s_q))$  is greater than  $\delta$ . The reduced flexibility score  $RF(i, p_q, s_q)$  measures how many units of flexible capacity are consumed by assigning appointment time  $q$  to patient  $i$ . Algorithm 8 describes how the reduced flexibility score is calculated.

---

**Algorithm 8** Reduced flexibility score

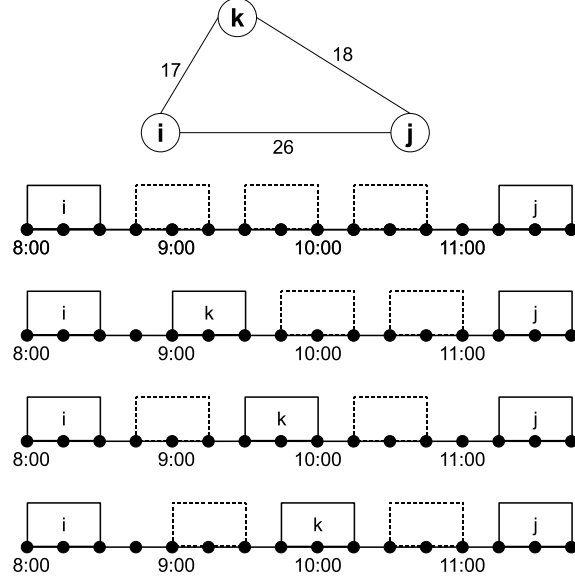
---

```

1: if ( $FC(p_q, s_q) > 0$ ) and ( $VC(p_q, s_q) > 1$ ) then
2:   if ( $VC(p_q, q) == 0$ ) and ( $h(o(p_q), i) > \delta$ ) then
3:      $RF(i, p_q, s_q) = \left\lceil \frac{h(o(p_q), i)}{\delta} \right\rceil$ 
4:   end if
5:   if ( $VC(q, s_q) == 0$ ) and ( $h(i, o(s_q)) > \delta$ ) then
6:      $RF(i, p_q, s_q) = \frac{h(i, o(s_q))}{\delta}$ 
7:   end if
8: end if

```

---



**Figure 10:** Example illustrating flexible schedule capacity

Consider the example shown in Figure 10. This is similar to the example in Figure 9, except that the appointment time for  $j$  is now 11:15 instead of 11:00. Thus, the visit capacity between  $i$  and  $j$  is 3, and there is one flexible capacity unit. The distance heuristic would select 9:00 for  $k$ . The associated reduced visit capacity would be zero, but the flexible unit would be consumed between  $i$  and  $k$  because the travel between  $i$  and  $k$  is greater than  $\delta$ . Choosing appointment time 9:30 or 9:45 would not reduce the visit capacity or consume the flexible unit. If 9:30 is selected, the flexible unit will be preserved for future use between  $k$  and  $j$ , while the flexible unit will be preserved between  $i$  and  $k$  if 9:45 is selected. In this paper, the convention we use to select between such alternatives is to append the hierarchical criteria of the distance heuristic. Within a given interval, this will have the effect of keeping the unit of flexible capacity between  $k$  and the interval endpoint which requires the most travel. In this example, the capacity heuristic would select 9:30.

Let  $\mathcal{S}_i(k)$  contain the appointment times which can feasibly be used for regular visits to patient  $i$ ,  $\mathcal{S}_i(k) = \{q : (q \in \mathcal{Q}_i^r) \text{ AND } (dOW(q) = k)\}$ . The hierarchical approach used to select the best appointment time for patient  $i$  on day  $k$  from the set of alternatives  $\mathcal{S}_i(k)$  is as follows.



1.  $\mathcal{S}_i^I(k) = \{q : RVC(i, q) = \min\{RVC(i, q') : q' \in \mathcal{S}_i(k)\}\}$
2.  $\mathcal{S}_i^{II}(k) = \{q : RF(i, p_q, s_q) = \min\{RF(i, p_{q'}, s_{q'}) : q' \in \mathcal{S}_i^I(k)\}\}$
3.  $\mathcal{S}_i^{III}(k) = \{q : c_r^{DH}(i, q) = \min\{c_r^{DH}(i, q') : q' \in \mathcal{S}_i^{II}(k)\}\}$
4.  $q^*(k) = \text{distanceTieBreak}(\mathcal{S}_i^{III}(k))$

### 3.6.3 Daily workload capacity

A similar hierarchical approach is used when selecting a patient's best appointment time combination  $O^*$ , which uses the appointment times  $q^*(k)$  for each  $k$  in the best visit day combination  $v^*$ . To additionally preserve capacity for future arriving patients which require multiple visits per week the approach uses a score denoted workload capacity  $WC(k)$  for each weekday  $k$ . Briefly, the workload capacity score mechanism attempts to balance nurse workload by weekday by selecting visit days with the fewest currently assigned patient visits. Without this mechanism, the nurse schedules may quickly become “full” for some weekdays while remaining relatively empty on others. For example, suppose the nurse has no available capacity on Monday, Wednesday, and Friday, but is completely free, having no patient appointments scheduled, on Tuesday and Thursday. If all currently revealed patient requests require three visits per week, the capacity on Tuesday and Thursday can not be utilized.

The **workload capacity** score of weekday  $k$  is defined as the number of patient visits that are currently assigned to day  $k$ . Let  $I_q$  be an indicator variable which is equal to 1 if  $o(q) \neq 0$ . The workload capacity score of an appointment time combination is the sum of the workload capacity scores of the days in the combination. The single weekday score,  $WC(k)$ , and the appointment time combination score,  $WC(v)$ , are calculated as described in Equations (52) and (53):

$$WC(k) = \sum_{q \in \mathcal{Q}_i^r : dOW(q)=k} I_q, \quad (52)$$

$$WC(v) = \sum_{q_k \in O} WC(k). \quad (53)$$

The full hierarchy used to select the appointment time combination for patient  $i$  in the capacity heuristic is now presented. Note that in each case, the score for the appointment time combination is a sum of the scores associated with each appointment time  $q^*(k)$  in the combination. Let  $\mathcal{S}_i$  contain the feasible appointment time combinations  $O \in \mathcal{O}_i$  for which the selected appointment times on the included days  $k$  represent the best choices on those days,  $\mathcal{S}_i = \{O = (q_1, \dots, q_{f_i}) : (O \in \mathcal{O}_i) \text{ AND } (q_j = q^*(k) \text{ if } dOW(q_j) = k, \forall j = 1, \dots, f_i)\}$ . The hierarchical approach used to select the best appointment time combination  $O^*$  for patient  $i$  from the set of alternatives  $\mathcal{S}_i$  is as follows.

1.  $\mathcal{S}_i^I = \left\{O : \sum_{q_j \in O} RVC(i, q_j) = \min\{\sum_{q_j \in O'} RVC(i, q_j) : O' \in \mathcal{S}_i\}\right\}$
2.  $\mathcal{S}_i^{II} = \left\{O : \sum_{q_j \in O} RF(i, p_{q_j}, s_{q_j}) = \min\{\sum_{q_j \in O'} RF(i, p_{q_j}, s_{q_j}) : O' \in \mathcal{S}_i^I\}\right\}$
3.  $\mathcal{S}_i^{III} = \{O : WC(O) = \min\{WC(O') : O' \in \mathcal{S}_i^{II}\}\}$
4.  $\mathcal{S}_i^{IV} = \left\{O : \sum_{q_j \in O} c_r^{DH}(i, q_j) = \min\{\sum_{q_j \in O'} c_r^{DH}(i, q_j) : O' \in \mathcal{S}_i^{III}\}\right\}$
5.  $O^* = \arg \min\{O \in \mathcal{S}_i^{IV}\}$

Once the best admission appointment time and appointment time combination have been identified for each new patient in the planning interval, the same hierarchy is used once again to determine in which order to admit the patients. Let  $\mathcal{S}$  in this case contain those newly arrived patients  $i \in \mathcal{P}_g$  for which feasible insertion options were found. Let  $O^*(i)$  denote the best appointment time combination for patient  $i$  and  $q_0^*(i)$  denote the best admission appointment for patient  $i$ . The hierarchical approach used to select the best patient  $i$  to admit from the set of alternatives  $\mathcal{S}$  is as follows.

1. Let  $RVC^* = \min_{i \in \mathcal{S}} \{RVC(i, q_0^*(i)) + \sum_{q_j \in O^*(i)} RVC(i, q_j)\}$   
 $\mathcal{S}^I = \left\{i : RVC(i, q_0^*(i)) + \sum_{q_j \in O^*(i)} RVC(i, q_j) = RVC^*\right\}$
2. Let  $RF^* = \min_{i \in \mathcal{S}^I} \{RF(i, p_{q_0^*(i)}, s_{q_0^*(i)}) + \sum_{q_j \in O^*(i)} RF(i, p_{q_j}, s_{q_j})\}$   
 $\mathcal{S}^{II} = \left\{i : RF(i, p_{q_0^*(i)}, s_{q_0^*(i)}) + \sum_{q_j \in O^*(i)} RF(i, p_{q_j}, s_{q_j}) = RF^*\right\}$

3. Let  $WC^* = \min_{i \in \mathcal{S}^{II}} \{WC(O^*(i))\}$

$$\mathcal{S}^{III} = \{i : WC(O^*(i)) = WC^*\}$$

4. Let  $c^{D*} = \min_{i \in \mathcal{S}^{III}} \{c^{DH}(i, q_0^*(i)) + \sum_{q_j \in O^*(i)} c_r^{DH}(i, q_j)\}$

$$\mathcal{S}^{IV} = \left\{i : c^{DH}(i, q_0^*(i)) + \sum_{q_j \in O^*(i)} c_r^{DH}(i, q_j) = c^{D*}\right\}$$

5.  $i^* = \arg \min_i \{i \in \mathcal{S}^{IV}\}$

### 3.7 Experiments and results

We perform a computational study on a set of test problems to compare the quality of schedules developed with the distance and capacity heuristics. In the study, we assume the nurse is available Monday through Friday each week ( $\mathcal{K} = \{1, 2, 3, 4, 5\}$ ) and has the appointment time menu  $\mathcal{M} = \{8:00, 8:15, \dots, 4:30\}$  with spacing of  $\delta = 15$  minutes available for scheduling patient visits each day. We assume all patients have a service duration of  $\Delta = 4$  weeks, which necessitates a planning horizon of 25 days. Planning is performed twice per day, at 7 a.m. and 12 p.m. All patients requesting service during the weekend are processed during the Monday morning planning interval.

The patient requests for each problem instance are generated such that  $\alpha$ , the expected number of patient requests per day, is homogenous throughout the study period. We experiment with  $\alpha = \{0.5, 0.7, \dots, 1.5\}$  patient requests per day. Patient  $i$  arriving in planning interval  $g$  has request arrival time  $a_i$  drawn uniformly at random from the interval  $(\text{time}(d_{g-1}, t_{g-1}), \text{time}(d_g, t_g)]$ , and weekly visit frequency  $f_i$  equal to 1, 2, or 3 visits per week with probabilities 0.05, 0.35, and 0.60. We assume each patient requires visits of length  $w = 30$  minutes, and that the set of allowable appointment time combinations for each patient  $i$  includes every possible  $f_i$ -day combination of the five-day work week.

The locations that comprise set  $\mathcal{N}$  from which patient requests can occur lie within a square geographic region subdivided into 900 equally-sized square subregions. The travel time  $h$  along one side of the region is 30 minutes for a small region test case and 60 minutes for a medium region test case. Three request location distributions, from which patient request locations are randomly drawn, are studied: uniform (U), clustered (C), and

combination (UC). Under location distribution U, each  $n \in \mathcal{N}$  generates a new request in planning interval  $g$  with probability  $\frac{\alpha}{2|\mathcal{N}|}$ . Under location distribution C, patient requests can only originate from locations falling within three small clusters inside the overall region. Under location distribution UC, patient requests are three times as likely to originate within five small clusters compared with elsewhere in the overall region. For each distribution, a total of  $\alpha$  requests are expected from  $\mathcal{N}$  in each planning interval  $g$ .

The total length of the study period is one year. A warmup period of four weeks is used, during which patient arrival rates are slowly increased to  $\alpha$ . A cooldown period of four weeks is also used, to allow all patients' episodes of care to end. The first eight weeks and last eight weeks are excluded from consideration when calculating performance metrics. The following metrics are used to compare the quality of the schedules developed.

- Visits per day: average number of patient visits performed by the nurse each day
- Visit acceptance rate:  $\frac{\text{number of patient visits scheduled}}{\text{number of patient visits requested}}$
- Travel time per visit: average amount of travel time required per patient visit

For each combination of experimental factors, Table 5 reports the observed value of the visits per day and visit acceptance rate metrics for the schedules generated using the capacity heuristic under column heading CH and the distance heuristic under column heading DH. The column under the “%” heading reports the percent change in performance from the distance heuristic to the capacity heuristic; positive values denote that CH improves over DH. The observed value of the travel time per visit metric is reported for each combination of experimental factors in Table 6 using the same column headings. Here, a positive value in the percent change column denotes that DH outperforms CH on this metric.

The computation times per planning interval in all problem instances were less than a minute, indicating that this approach could be implemented in a real-time patient scheduling environment. In most problem instances, we see that the capacity heuristic outperforms the distance heuristic on the primary metrics of visits per day and visit acceptance rate, and the distance heuristic outperforms the capacity heuristic on the metric of average travel per

patient visit. This is to be expected, as the capacity heuristic focuses on preserving capacity for admitting patients, and the distance heuristic focuses solely on minimizing distance.

### **3.7.1 Absolute performance of the capacity heuristic**

The problem instances which correspond most closely with a realistic patient volume for a single nurse are those where  $\alpha \in \{0.9, 1.1\}$  patient requests per day. If 100% of the patient requests are able to be accommodated, this number of patient requests would result in a daily visit volume equal to the daily capacity of the nurse. Of course, many factors can contribute to the inability to achieve near 100% acceptance rates; primarily problem dynamics, and the level of inflexibility present in this scheduling problem. The maximum visit acceptance rates achieved in instances having arrival rates of 0.9 and 1.1 patient requests per day are 79.7% and 71.8%, respectively. This occurs when the region size is small and patient requests arrive according to the combination distribution. The nurse productivity in these instances is 8.11 and 8.77 visits per day. These numbers will serve as a baseline as we discuss the impact of changing the arrival rate, region size, and location distribution.

#### **Impact of patient request arrival rate**

As request arrival rate increases within the range  $\{0.5, \dots, 1.5\}$ , nurse productivity increases and visit acceptance rate decreases. At the lowest request arrival rate, the visit acceptance rate is 94%, averaged across all location distributions in the small region. Nurse productivity is limited to 5.5 visits per day. Note that the nurse has capacity to perform at most 12 patient visits per day in this problem setting. The volume of new patient requests at  $\alpha = 0.5$  is not sufficient to consume the capacity available in the nurse's schedule. The large quantity of excess capacity in the nurse's schedule allows the heuristic to accommodate almost all of the arriving requests, regardless of the travel per visit required to do so. The opposite extreme occurs at the highest request arrival rate. The capacity heuristic is able to schedule 9.35 visits per day when  $\alpha = 1.5$ , but the visit acceptance rate is limited to 54.5%. With this volume of request arrivals, the heuristic selects only those requests which are most desirable from a nurse productivity standpoint.

#### **Impact of region size**

**Table 3:** Capacity heuristic output for  $\alpha = 1.5$ , location distribution U

	small	medium
Visits per day	9.37	7.74
Length of each visit	30	30
Travel time per visit	10.102	19.144
Visit time per day	281.1	232.3
Travel time per day	94.7	148.2
Total travel plus visit time per day	375.7	380.6
Idle time per day	134.2	129.6

We can expect that visits per day and visit acceptance rates will decrease as the size of the region increases, all other problem parameters being held constant. As the region being served by a single nurse becomes larger, a larger portion of his available time will be spent traveling between patient visits. Depending on problem parameters, the schedules generated by the capacity heuristic for the medium region instances have 0.19 to 1.63 fewer visits per day, on average, than for the small region. The smallest difference, 0.19 visits per day, occurs when  $\alpha = 0.5$  under the clustered distribution. In this problem instance, the nurse's schedule is only at 50% capacity, so higher travel times minimally impact the ability of the nurse to admit new patients. The largest difference, 1.63 fewer visits per day, occurs when  $\alpha = 1.5$  under the uniform distribution. The output from these instances can be used to calculate the total visit time and travel time required to complete the average number of visits per day on the small and medium regions. This information is displayed in Table 3. Visiting 7.74 patients per day in the medium region requires the same amount of time as visiting 9.37 patients per day in the small region, because the average travel time per visit is almost doubled as the size of the region is doubled.

### Impact of location distribution

The type of location distribution for which the capacity heuristic yields the best results is different between the small and medium regions. Table 4 displays the visits per day achieved by the capacity heuristic in both region sizes on all location distributions, averaged across arrival rates. On the small region, the clustered distribution tends to yield the *lowest* nurse productivity when compared to the uniform and combination distributions. On the medium region, the clustered distribution tends to yield the *highest* nurse productivity. This should be attributed to the characteristics of the instances being solved, not to the mechanics of the

**Table 4:** Visits per day, averaged across all arrival rates

heuristic	region	U	C	UC
capacity	small	8.08	7.70	8.04
	medium	6.56	7.36	6.58
distance	small	7.70	7.63	7.77
	medium	6.58	7.06	6.64

capacity heuristic, because the results obtained via the distance heuristic follow a similar pattern. In this problem setting, the highest potential for maximizing nurse productivity occurs when the travel between pairs of consecutive patient visits consumes no more than one appointment time, *i.e.*, the travel segment is less than 15 minutes ( $\delta$ ) in length. On the small region, the clustered distribution has the lowest likelihood of any two patients being within 15 minutes travel of each other. On the medium region, the clustered distribution has the highest such likelihood.

### 3.7.2 Relative performance of the capacity and distance heuristics

The capacity heuristic outperforms the distance heuristic on the metrics of visits per day and visit acceptance rate in most problem instances. The average percent improvement is 4% over all improving instances. In those instances in which the distance heuristic performs better, the visits per day and visit acceptance rates are 2.8% lower on average when the capacity heuristic is used. On the metric of travel per patient visit, the distance heuristic always outperforms the capacity heuristic. The travel times required by the capacity heuristic are 8.7% higher on average than those required by the distance heuristic.

The discussion of the results will first focus on the metrics of visits per day and visit acceptance rate, because the primary problem objective is to maximize the number of patients served. The discussion of the results on the clustered distribution instances is kept separate from the uniform and combination distribution instances because the trends are different.

#### Uniform and combination location distributions

On the small region, the capacity heuristic always achieves higher nurse productivity and visit acceptance rates than the distance heuristic for the uniform and combination location distributions. As  $\alpha$  increases, the magnitude of the percent improvement decreases, until

the capacity heuristic is just slightly outperforming the distance heuristic at the highest request arrival rate. It is reasonable for the capacity and distance heuristics to perform equally well at high request arrival rates. In such scenarios, both heuristics admit only half of the patients that are requesting service. Both heuristics are admitting only those patients which are most desirable from a nurse productivity standpoint.

As the size of the region increases from small to medium on these two location distributions, the magnitude of the percent improvement of the capacity heuristic over the distance heuristic decreases. In problem instances with increased arrival rate *and* increased region size, the distance heuristic outperforms the capacity heuristic. In medium region size instances where  $\alpha > 0.9$ , the capacity heuristic achieves 2.8% *fewer* visits per day than the distance heuristic. Perhaps in these instances, the capacity heuristic is being too optimistic in its selection of appointment times for arriving patients. Any time that a given patient insertion will induce a travel segment greater than 15 minutes in length to the predecessor or successor, the capacity heuristic attempts to reserve enough time between the two patient visits to insert an additional future arriving patient. Ideally, a future patient will arrive that enables separating the longer travel segment into two segments, each less than 15 minutes in length. However, if such a patient is not found, the time that was reserved in anticipation of the future request will become idle time in the nurse's schedule. The distance heuristic would have accepted the longer travel segment and scheduled the patient adjacent to the interval endpoint, without reserving the additional time that is ultimately left idle.

### **Clustered distribution**

The trends in relative performance of the two heuristics are quite different for the clustered location distribution. The magnitude of the percent improvement of the capacity heuristic over the distance heuristic still decreases as the arrival rate increases. However, the trend across the change in region size is reversed. The percent improvement of the capacity heuristic over the distance heuristic *increases* as the region size increases. The capacity heuristic always outperforms the distance heuristic on the medium region, by an average of 4.6% across all arrival rates. On the small region, the capacity heuristic outperforms the distance heuristic at arrival rates below 1.3 requests per day, but the distance



heuristic performs 2.7% better at the higher arrival rates.

### **Average travel per visit**

The distance heuristic always outperforms the capacity heuristic on average travel per visit. The capacity heuristic generates solutions which require 9.34% more travel time per patient visit on average across all instances. This is to be expected, as the sole objective of the distance heuristic is to minimize total travel distance, while that is a lower level objective of the capacity heuristic. The travel per visit required by the capacity heuristic is as much as 15% higher in one instance, but that represents an absolute performance difference of only 3 minutes per visit. In this problem, minimizing travel distance is an important consideration, but the primary objective is maximizing the number of patient requests that are served. The distance heuristic is not able to take advantage of travel time savings to perform additional patient visits.

### ***3.8 Conclusions and future work***

These results indicate that the capacity heuristic is an effective scheduling approach for dynamic periodic home health nurse scheduling problems on a wide variety of patient request location distributions. Table 7 summarizes which heuristic performed best in each of the parameter combinations tested. The capacity heuristic achieved better performance on the most important metric of visits per day than the distance heuristic in most problem instances. The instances in which the distance heuristic performed better were problem settings less likely to be encountered in practice. Either the request arrival rate was too high, or the region size was too large, to expect a single nurse to serve all patient requests originating within the region. Due to fast computation times, either approach could be implemented in a real-time patient scheduling environment.

Home health agencies typically assign staff to cover a predetermined geographic region. Thus, managers can compare their clinician service regions and historical patient demand patterns to Table 7 to see which approach is more appropriate for their organization. In most metropolitan areas, nurse service regions which are 30 minutes or less in diameter should generate a sufficiently high volume of patient visits for a single nurse. Thus, the capacity

**Table 5:** Visits per day and visit acceptance rates

Metric	Distribution	$\frac{\alpha}{2}$	small ( $h = 30$ )			medium ( $h = 60$ )		
			CH	DH	%	CH	DH	%
Vis./ day	U	0.25	5.634	5.139	9.622	5.08	4.802	5.792
		0.35	7.005	6.49	7.932	5.954	5.749	3.557
		0.45	8.269	7.795	6.081	6.644	6.609	0.522
		0.55	8.844	8.479	4.31	7.009	7.101	-1.291
		0.65	9.172	8.944	2.552	7.188	7.48	-3.901
		0.75	9.576	9.37	2.205	7.507	7.744	-3.061
	C	0.25	5.453	5.154	5.807	5.346	4.962	7.743
		0.35	6.935	6.691	3.642	6.703	6.217	7.814
		0.45	7.695	7.634	0.795	7.371	7.03	4.852
		0.55	8.507	8.221	3.485	7.917	7.663	3.312
		0.65	8.677	8.903	-2.544	8.317	8.144	2.122
		0.75	8.932	9.19	-2.805	8.488	8.35	1.65
	UC	0.25	5.445	5.202	4.674	5.007	4.769	4.997
		0.35	7.147	6.609	8.134	6.028	5.967	1.022
		0.45	8.111	7.8	3.986	6.685	6.679	0.091
		0.55	8.774	8.556	2.54	6.987	7.131	-2.028
		0.65	9.26	9.132	1.396	7.287	7.546	-3.444
		0.75	9.532	9.329	2.176	7.51	7.767	-3.306
Vis. acc rate	U	0.25	0.936	0.85	10.12	0.845	0.802	5.336
		0.35	0.876	0.813	7.709	0.743	0.717	3.528
		0.45	0.787	0.743	5.869	0.633	0.632	0.127
		0.55	0.687	0.659	4.262	0.548	0.551	-0.599
		0.65	0.628	0.611	2.816	0.49	0.511	-4.187
		0.75	0.562	0.547	2.631	0.439	0.455	-3.56
	C	0.25	0.948	0.896	5.813	0.929	0.862	7.746
		0.35	0.88	0.846	3.995	0.85	0.79	7.61
		0.45	0.774	0.766	1.058	0.741	0.707	4.897
		0.55	0.695	0.671	3.637	0.65	0.627	3.62
		0.65	0.568	0.582	-2.456	0.545	0.533	2.251
		0.75	0.518	0.528	-1.895	0.49	0.48	1.894
	UC	0.25	0.931	0.888	4.841	0.852	0.81	5.123
		0.35	0.886	0.816	8.642	0.744	0.737	1.018
		0.45	0.797	0.763	4.376	0.657	0.655	0.183
		0.55	0.718	0.7	2.63	0.568	0.581	-2.34
		0.65	0.611	0.603	1.21	0.477	0.5	-4.579
		0.75	0.564	0.552	2.101	0.443	0.459	-3.595

**Table 6:** Average travel time per visit

Metric	Distribution	$\frac{\alpha}{2}$	small ( $h = 30$ )			medium ( $h = 60$ )		
			CH	DH	%	CH	DH	%
Avg trav ratio	U	0.25	11.465	10.241	11.95	22.534	19.824	13.668
		0.35	11.303	10.325	9.473	22.413	20.24	10.739
		0.45	11.03	10.179	8.362	21.717	20.077	8.168
		0.55	10.743	9.938	8.108	21.002	19.494	7.734
		0.65	10.778	10.093	6.782	20.994	19.72	6.463
		0.75	10.598	10.102	4.914	20.266	19.144	5.861
	C	0.25	11.425	10.336	10.538	21.237	18.426	15.258
		0.35	11.081	9.97	11.142	19.698	17.621	11.79
		0.45	10.898	9.587	13.672	18.971	17.013	11.512
		0.55	10.712	9.659	10.897	18.122	16.952	6.906
		0.65	10.64	9.57	11.18	17.421	16.952	2.766
		0.75	10.576	9.664	9.436	17.506	16.906	3.551
	UC	0.25	11.653	10.515	10.83	23.281	20.928	11.244
		0.35	11.222	10.216	9.842	22.084	20.882	5.757
		0.45	11.078	10.288	7.685	21.553	20.089	7.286
		0.55	11.078	10.08	9.9	21.761	20.108	8.217
		0.65	10.953	10.212	7.253	21.114	19.708	7.132
		0.75	10.872	10.248	6.093	20.485	20.046	2.189

**Table 7:** Recommendations for using the capacity and distance heuristics

Location distribution	Region size	Daily request arrival rate $\alpha$					
		0.5	0.7	0.9	1.1	1.3	1.5
Uniform and Combination	small ( $h = 30$ min.)	C					
	medium ( $h = 60$ min.)	C			C/D	D	
Clustered	small	C				D	
	medium	C					

heuristic should be used. The exception occurs when the volume of request arrivals is very high and patient locations follow the clustered distribution. Then, the distance heuristic is a better choice than the capacity heuristic. However, a better decision altogether would be to add staff to the region or decrease the size of the region such that the level of patient requests being served by a single nurse would be reasonable. Without doing so, both heuristics would only be able to accommodate approximately 50% of the patient requests.

Patient location distributions in suburbs and rural areas can sometimes necessitate that the service region of the nurse be as large as 60 minutes in diameter, similar to the medium region instances tested. If the service area is this large and patients are located according to the clustered distribution, the capacity heuristic should be used to schedule patient visits. If the patients are located according to uniform or combination distributions and the request arrival rate is lower than average, the capacity heuristic should still be used. However, if request arrival rates are high under those two distributions, the distance heuristic is recommended.

The dynamic periodic home health nurse scheduling problem studied in this paper is very rigid. Patient appointment times cannot be rearranged to accommodate new patient requests once their visit schedule has been established. An additional area for future research is to determine the value of the capacity heuristic in a more flexible periodic problem setting. Perhaps it would be possible to allow a patient's appointment time to vary by 15 minutes each week, or 30 minutes each week. Specific research questions that could be addressed are (i) what would be gained in terms of nurse productivity, and (ii) would the capacity approach still be effective.

## CHAPTER IV

### HOME HEALTH NURSE DISTRICTING

#### 4.1 *Introduction*

Home health agencies provide health care services to patients living within service regions that often span over 5,000 square miles in large metropolitan areas. The services are delivered to patient locations using a staff of nurses that visit patients in their homes. Chapter 3 describes an operational planning problem encountered by home health agencies each day: develop visit schedules for newly arrived patient requests, specifying which nurse will perform the visits to each patient and the days and times those visits will occur. The capacity of each nurse is limited, and the productivity of each nurse is influenced by the size of the region in which their assigned patient requests are distributed. Thus, a tactical planning problem that in part determines the quality of the solutions that can be obtained for the operational planning problem is to divide the service region into subregions, or districts, to be served by each nurse.

We define home health nurse districting (HHND) problems for a connected service region that includes a set of subunits, *e.g.*, zip codes, and a staff of nurses with limited capacities that must be deployed to serve patient demand within the region. Nurse service districts must be designed such that each zip code is assigned to exactly one district, and the workload of each district must be within the allowable workload bounds of the nurse or team of nurses serving the district. To limit nurse travel between patient visits, districts should be geographically compact and contiguous. Additionally, the number of nurses serving each district should be small so that the number of different nurses any patient may receive visits from during their episode of care is limited. Consistency of service provider, referred to in the health literature as continuity of care, has positive implications for care outcomes. Studies have shown a correlation between continuity of care, increased patient satisfaction, and decreased hospitalizations and emergency room visits [13]. Although we are not aware of a

study directly linking improved home care outcomes with continuity of care, the correlation is expected because correlation between continuity of care and improved health outcomes is most consistently indicated for patients with chronic conditions, which comprise the majority of patients receiving home care services.

In this chapter, we present a set partitioning model and optimization-based heuristic for the home health nurse districting problem. The goal of the approach is to create home health nurse districts which are contiguous, compact, balance the expected workload across districts, and minimize expected operational routing and scheduling costs. In Section 4.2, we review literature on related problems. Sections 4.4.4 and 4.5.3 define the problem and mathematical formulation, and Section 4.5 describes our solution approach. The results of a computational experiment are given in Section 4.6 and conclusions are given in Section 4.7.

## 4.2 *Literature review*

Districting problems have appeared in the literature in a variety of applications, for example: police officer territories, [20], sales territories [27, 45], school districts [14], vehicle delivery districts [25], and political districts [11, 24, 28, 29, 35, 39]. In each of the above studies, a set of subunits must be grouped into a number of contiguous districts such that each subunit is included in exactly one district and some set of side constraints are satisfied. The side constraints vary depending on the application, but typically include balancing attributes across districts and creating districts which are geographically compact. For example, political districting studies balance the population of eligible voters in each district and create districts which resemble “regular” shapes such as circles and squares. School districting studies balance the number of students assigned to each school and create districts which minimize the average distance students must travel to reach their assigned school.

Table 8 summarizes select districting applications from the literature, indicating the type of model used in each study, and indicating whether problem components such as attribute balancing and compactness are treated as problem objectives or constraints. Attribute balancing, which is of primary importance in many applications, is most often enforced

**Table 8:** Select districting applications from the literature

Application	Reference	Objective			Constraints			Model	
		Compactness	Balancing	Other	Compactness	Balancing	Other	Location-allocation	Set partitioning
Political districts	Garfinkel and Nemhauser [24]		x		x	x			x
	Hess and Samuels [27]	x				x		x	
	Hojati [29]	x				x		x	
	Mehrotra et al. [35]	x				x			x
	Ricca and Simeone [39]	x	x	x					x
	Bozkaya et al. [11]	x	x	x					x
Sales territories	Hess et al. [28]	x				x		x	
	Zoltners and Sinha [45]	x				x		x	
Police districts	D’Amico et al. [20]	x				x	x		x
School districts	Caro et al. [14]	x				x	x		

using constraints. Exact balance is often not achievable without violating other feasibility conditions (*e.g.*, splitting subunits between districts); thus, most approaches constrain the attribute value of each district to be within allowable bounds. With the exception of Garfinkel and Nemhauser [24], each study summarized uses an objective function which maximizes the total compactness of all districts. However, the measurement used for compactness varies; each study employs a metric both appropriate for the application and also easily integrated into the chosen solution method.

Select studies in Table 8 include additional objective function components and constraints, also specific to the application. For example, Bozkaya et al. [11] includes an objective function component that penalizes subunit-to-district assignments which differ from an existing plan, and D’Amico et al. [20] includes a constraint on patrol car response time.

The two primary mathematical programming formulations for districting problems found in the literature are location-allocation and set partitioning. When location-allocation models are used, it is assumed that a set of fixed district centers or a finite set of possible district centers is given. Required decisions include selecting which districts centers to open, if necessary, and assigning subunits to the selected district centers, subject to a set of constraints. Modeling the districting problem using set partitioning does not require that the center of a

district be specified. Instead, the set of all feasible districts is assumed to be available, and required decisions include selecting a subset of districts such that each subunit is included in exactly one district. Location-allocation and set partitioning models from the districting literature are now reviewed.

#### 4.2.1 Location-allocation models for districting problems

Hess et al. [28] is credited with the original location-allocation formulation for the political districting problem, which Hess and Samuels [27] later adapted for the sales territory alignment problem. In the formulation, there are a set of  $m$  district centers and  $n$  subunits. Each subunit is assigned to exactly one district center, such that some attribute is balanced across  $m$  districts, and the total cost of assigning subunits to district centers is minimized. Let  $c_{ij}$  denote the cost of assigning subunit  $i$  to district center  $j$ , and let  $x_{ij}$  be a binary variable indicating whether subunit  $i$  is assigned to district center  $j$ . Additionally, let  $a_i$  denote the attribute value for subunit  $i$ , and denote the average attribute value across  $m$  districts as  $\bar{a} = \frac{\sum_i a_i}{m}$ . The location-allocation formulation ( $LA$ ) is stated as follows:

$$\text{Minimize } \sum_{j=1}^m \sum_{i=1}^n c_{ij} x_{ij} \quad (54)$$

$$\text{Subject to } \sum_{i=1}^n a_i x_{ij} = \bar{a} \quad \forall j = 1, \dots, m \quad (55)$$

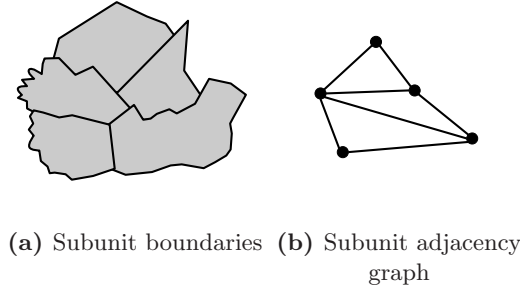
$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, n \quad (56)$$

$$x_{ij} \in \{0, 1\} \quad (57)$$

The objective function in (54) minimizes the total cost of assigning subunits to district centers, which is a function of the weighted Euclidean distance from each subunit to its assigned district center. Constraints (55) ensure attribute balancing across districts. As it is unlikely that exact balance of attributes can be achieved, typical location-allocation models for districting problems found in the literature replace the equality constraints with a set of lower and upper bound inequality constraints. Constraints (56) ensure each subunit is assigned to exactly one district.

Hess and Samuels [27] solve  $LA$  by relaxing the integrality constraints, solving the resultant LP, and rounding the solution so that subunits are not split between districts. A primary drawback of this approach is that the rounded solution may not be optimal, and is also likely to violate attribute balancing constraints. Additionally, district contiguity is encouraged by the objective function but is not guaranteed. Linear district contiguity constraints which can be incorporated in  $LA$  are given in Shirabe [41], but the author demonstrated through a computational study that problem instances containing more than 100 subunits are intractable.

Zoltners and Sinha [45] develop an alternate approach for incorporating contiguity constraints into a location-allocation formulation. An adjacency graph is created which contains nodes associated with each subunit and district center. The graph includes edges between pairs of subunits and district centers which are adjacent. A set of subunits and the associated adjacency graph is depicted in Figure 11.



**Figure 11:** Subunit boundary map and associated adjacency graph

Using the adjacency graph, hierarchical subunit adjacency trees rooted at each district center are developed, where the tree at each district center specifies the shortest path measured in travel time to each subunit along edges in subunit adjacency graph. Precedence constraints are derived from the trees, and specify that a subunit can only be assigned to a district if a subunit one level higher on the tree (*i.e.*, one step closer to the district center) is also assigned to the district center. The authors obtain a solution to the resultant model using Lagrangian relaxation on the precedence constraints.

Hojati [29] develops a facility location formulation that does not require a predefined



set of district centers, but instead allows  $m$  district centers to be selected from  $n$  population subunits. Letting  $y_j$  indicate whether subunit  $j$  is selected as a district center, the model is:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (58)$$

$$\text{Subject to } \sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1 \dots n \quad (59)$$

$$\sum_{i=1}^n a_i x_{ij} = \bar{a} y_j \quad \forall j = 1 \dots n \quad (60)$$

$$\sum_{j=1}^n y_j = m \quad (61)$$

$$x_{ij} \leq y_j \quad \forall i = 1 \dots n, j = 1 \dots n \quad (62)$$

$$0 \leq x_{ij} \leq 1 \quad \forall (i, j) \text{ pairs} \quad (63)$$

$$y_j \in \{0, 1\} \quad (64)$$

Like  $LA$ , this model does not guarantee contiguity. The differences between this model and  $LA$  are that: (i)  $m$  district centers must be selected (61), (ii) subunits can only be assigned to district centers that are selected (62), and (iii) subunits may be split between districts (63). The authors propose a rounding scheme to resolve those subunits which are split between districts.

#### 4.2.2 Set partitioning models for districting problems

Set partitioning formulations provide the flexibility to construct feasible districts and evaluate their cost *outside* of the core optimization problem. Therefore, such approaches can often handle complicating constraints (like contiguity requirements) on district structure and complex objective functions more easily.

Let  $\mathcal{J}$  denote the set of all feasible districts, and let  $\gamma_{ij}$  be equal to 1 if district  $j$  includes subunit  $i$  and 0 otherwise. Let  $C_j$  be the cost of district  $j$ , and let  $y_j$  be a binary variable indicating whether district  $j$  is selected. The set partitioning formulation for districting problems, denoted as  $SP$ , is stated as follows:

$$\text{Minimize } \sum_{j \in \mathcal{J}} C_j y_j \quad (65)$$

$$\text{Subject to } \sum_{j \in \mathcal{J}} \gamma_{ij} y_j = 1 \ \forall \ i = 1 \dots n \quad (66)$$

$$\sum_{j \in \mathcal{J}} y_j = m \quad (67)$$

$$y_j \in \{0, 1\} \quad (68)$$

The objective function minimizes the total cost of all selected districts. Constraints (66) ensure that each zip code is included in exactly one district. Constraint (67) ensures exactly  $m$  districts are selected. Both exact and approximate solution approaches for this formulation are found in the literature.

Garfinkel and Nemhauser [24] develops an enumerative approach for solving  $SP$  for a political districting problem. The set of all districts satisfying contiguity, shape compactness, distance compactness, and population compactness criteria are enumerated. Each of the compactness measures are specified as nonlinear functions of the subunit groupings. For example, shape compactness is defined as the ratio of the maximum distance between any two included subunits and the area of the district. Letting  $x_i$  be a binary variable indicating whether subunit  $i$  is included in district  $j$ , and letting  $d_j$  denote the maximum distance between any two subunits included in the district,  $d_j$  is computed using the following nonlinear function:

$$d_j = \max\{d_{ik} x_i x_k | i \in V, j \in V\}. \quad (69)$$

Given the set of feasible districts, the authors use  $SP$  to select a subset of  $m$  districts which cover each subunit exactly once, such that attribute “imbalance” across districts is minimized. Because the number of feasible districts is exponential in the number of subunits, the enumerative approach can be computationally unattractive. In the study performed by the authors, the approach worked well only in instances containing less than 40 subunits.

Mehrotra et al. [35] present an optimization based solution approach for solving  $SP$  which does not require enumerating all feasible districts in advance. They instead develop a branch-and-price procedure, using column generation to generate new districts on an as-needed basis. The procedure first solves the linear relaxation of  $SP$  on an initial subset of feasible districts. Then, a pricing subproblem which uses a linear cost function is used to find negative reduced costs districts which may improve the solution. Solving the pricing problem to optimality requires solving a two-sided knapsack problem with additional contiguity side constraints for each subunit, where each problem finds the minimum cost district centered at the associated subunit. While this approach is optimization-based, it is not exact, because the contiguity constraints the authors employ in the pricing subproblem exclude some portions of the feasible region.

Bozkaya et al. [11] develop a set partitioning model for the political districting problem that includes a contiguity constraint and an objective function which incorporates various criteria with nonlinear representations: population equality, compactness, socio-economic homogeneity, similarity to the existing plan, and integrity (non-splitting) of communities. They solve the model using a tabu search procedure that includes two neighborhoods. The first is a shift neighborhood, which includes all feasible solutions that can be reached from the current solution by moving a subunit from its currently assigned district to another district. The second is a swap neighborhood, which includes all feasible solutions that can be reached by swapping a pair of subunits between a pair of districts.

Ricca and Simeone [39] also develop a heuristic procedure to solve a set partitioning formulation of the political districting problem. Four heuristic methods (descent, tabu search, simulated annealing, old bachelor acceptance) which use shift neighborhoods are evaluated under a variety of objective functions (population equality, compactness, administrative conformity). Excluding descent, each method demonstrated good performance on their test problem.

Each of the above political districting studies developed approaches to group subunits into contiguous and compact districts such that each subunit is assigned to exactly one district and voter population count is balanced across districts. The primary differences

between these applications and the home health nurse districting problem we study are as follows.

- The attribute balanced across districts cannot be represented as a simple count of the patients in each district. The objective is to balance nurse workload, which includes time spent visiting patients and time spent traveling between patient locations. Home health agencies often serve geographic regions comprised of both metropolitan and rural areas, where travel between patient locations is expected to be higher in the less densely populated portions of the region.
- The compactness measures used in the political districting literature favor districts of regular shape. The district cost measure we adopt favors districts which maximize the productivity of a mobile workforce by minimizing the travel required to serve patient demand.

Accurately modeling these problem characteristics requires nonlinear representations of both district cost and district workload balance constraints.

### 4.3 *Problem statement*

We now state the home health nurse districting problem (HHND). In this study, we assume that demand information is available at the zip code level. The approach is easily extended to cases where a different level of aggregation is more appropriate.

**Service region:** There is a connected service region which contains a set of  $\mathcal{N} = \{1, \dots, n\}$  zip codes, where each zip code has area  $v_i$  and demand  $\rho_i$ , measured in the number of visits required to patient locations in zip code  $i$  per day. Visits to patients in zip code  $i$  have duration  $w_i$ . The specific locations to which visits are required each day are not known in advance, but are assumed to be independent and uniformly distributed throughout the zip code with density  $\frac{\rho_i}{v_i}$ .

**Nurses:** A set of  $k$  nurses is available each day to serve patient demand within the service region. Each nurse has a daily workload target  $b$ , where nurse workload is measured in hours and includes time spent visiting patients and traveling between patient locations. Nurses

work in teams of size  $f$ ; thus the target daily workload of a single team of nurses is  $\beta = bf$ .

The problem is to develop  $m = \frac{k}{f} \in \mathbb{Z}^+$  contiguous districts, one for each team of nurses, such that the daily workload is balanced across districts. The daily workload of a district is defined as the time required for  $f$  nurses to serve the demand of each zip code comprising the district, including time spent visiting patients and traveling between patient visits. Workload is balanced if the workload of each district served by  $f$  nurses is within an allowable percent deviation from the target workload  $\beta$  specified by parameter  $\alpha$ , where  $0 \leq \alpha \leq 1$ . Smaller values of  $\alpha$  allow less disparity in district workload.

The objective is to minimize the total cost of all districts, where the cost of a district is defined as the total travel time required for each of  $f$  nurses to serve an equal fraction of demand within the district.

#### 4.4 Mathematical model

A set partitioning model is developed for HHND. Given a set of contiguous districts which are feasible with respect to workload bounds, the model selects  $m$  districts which cover each zip code exactly once and minimize total cost. District demand, cost, and feasibility are formally defined prior to specifying the model.

##### 4.4.1 District demand

The daily demand of a district is equal to the sum of the demands of each included zip code. Let  $\gamma_{ij}$  be equal to 1 if zip code  $i$  is included in district  $j$  and 0 otherwise. Let  $\lambda_j$  denote district demand, calculated as follows:

$$\lambda_j = \sum_{i=1}^n \gamma_{ij} \rho_i. \quad (70)$$

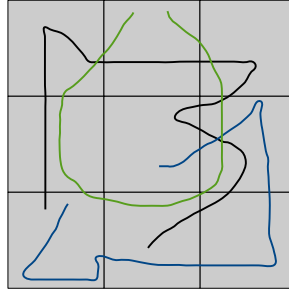
##### 4.4.2 District cost

We define district cost to be the expected daily routing costs the nurse or team of nurses assigned to a district will incur serving demand originating within the district. Because a primary objective is balancing workload, we assume that each nurse assigned to a district serves an equal portion of demand within the district, visiting  $\frac{\lambda_j}{f}$  patient locations each day.

We assume that the geographic distribution of the locations visited by each nurse is uniform with density  $\frac{\lambda_j}{fV_j}$ , where  $V_j$  denotes the area of the district and is calculated as follows:

$$V_j = \sum_{i=1}^n \gamma_{ij} a_i. \quad (71)$$

Consider the daily activity of each nurse. At the beginning of the day, the nurse leaves his home, visits  $\frac{\lambda_j}{f}$  uniformly distributed points in  $j$ , and returns home after the last patient is visited. If the patient locations are visited in order of the minimum duration TSP tour, district cost can be estimated as the cost of the TSP through the nurse's home location and  $\frac{\lambda_j}{f}$  uniformly distributed points in district  $j$ . If the travel between the nurse's home and the first and last patient locations is excluded from consideration, the estimate reduces to the cost of the Hamiltonian path through  $\lambda_j$  uniformly distributed points in  $j$ . Figure 12 depicts possible travel for three nurses serving a district on a given day.



**Figure 12:** Daily nurse travel as Hamiltonian paths through random points in district

In this study, we adopt two methods for approximating the cost of the Hamiltonian path traversed by each nurse in his assigned district. The first method approximates the distance between patient locations included in the path using an asymptotic formula for the length of the expected TSP tour through a number of randomly selected points in a connected area of known size. The second method approximates the interpoint distance as the average separation between actual patient locations included in the district. Using each method, the total cost of a district is the product of the approximated interpoint distance, the number of arcs traversed by each nurse  $\left(\frac{\lambda_j}{f} - 1\right)$ , and the number of nurses serving the district. The two methods are referred to respectively as *TSP* and *SEP* with corresponding

district costs denoted  $C_j^{TSP}$  and  $C_j^{SEP}$ .

### **TSP approximation**

A well-established result in the routing literature [5] states that the estimated cost of a TSP serving  $N$  customers which are independently scattered in a connected area of size  $A$  following a uniform distribution can be estimated as:

$$c(TSP) = k\sqrt{NA}. \quad (72)$$

In Equation (72),  $k$  is a constant that differs based on the distance metric being used. Much research has focused on finding accurate estimates for  $k$  as the number of points being visited goes to infinity. The most recent estimates give  $k$  as 0.7214 for the Euclidean metric [33]. We can first develop a district cost estimate for the case where a single nurse serves each district by replacing  $A$  with  $V_j$  and  $N$  with  $\lambda_j$  in Equation (72). The TSP estimate is multiplied by  $(1 - \frac{1}{\lambda_j})$  to obtain an estimate for the Hamiltonian path distance:

$$C_j^{TSP} = 0.7214 \left(1 - \frac{1}{\lambda_j}\right) \sqrt{\lambda_j V_j}. \quad (73)$$

To extend this estimate for the case where a district is served by a team of  $f$  nurses visiting  $\frac{\lambda_j}{f}$  locations each, we retain the assumption that the locations served by a single nurse are independent and uniformly scattered throughout the entire district. Because the home health model favors continuity of the patient-nurse relationship throughout a patient's episode of care, it is unlikely that the daily activities of the nurse are contained within a subregion of size  $\frac{V_j}{f}$  within the district. The revised district cost estimate is given in Equation (74), which, for  $f = 1$ , reduces to Equation (73):

$$C_j^{TSP} = 0.7214 f \left(1 - \frac{f}{\lambda_j}\right) \sqrt{\frac{\lambda_j}{f} V_j}. \quad (74)$$

The routing cost estimate given here may seem optimistic; we have shown in Chapters 2 and 3 that patient locations are not always visited using minimum-distance tours in solutions to HHNRS problems. However, recall also that in the periodic problem, there is some opportunity to assign visits to patient locations which are in close proximity to each other

to the same visit day. In this sense, Equation (74) may seem pessimistic, because it assumes patient locations are scattered uniformly through the entire district. In applications where patient locations are not assumed to be independent and uniformly distributed throughout each district, an appropriate method for estimating district routing costs should be selected. For example, a generalized approach for estimating routing cost that accounts for spatial variations in demand densities is developed in Blumenfeld and Beckmann [10].

#### ***SEP* approximation**

District cost approximation method *SEP* makes use of data regarding actual patient locations in the geographic service area; thus the assumption of uniformly distributed patient locations is not required. We assume that the number of daily visits to each patient location is homogenous throughout the service region. Denote the average separation between patient locations included in district  $j$  as  $\bar{h}_j$ . Then, district cost is estimated as:

$$C_j^{SEP} = f\bar{h}_j \left( \frac{\lambda_j}{f} - 1 \right). \quad (75)$$

#### **4.4.3 District workload feasibility**

In order for a district to be feasible, it should be contiguous, and the expected daily workload should be within the allowable bounds of the nurse or team of nurses assigned to serve the district. The expected daily workload of a district, denoted as  $T_j$ , includes the time spent in the patients' homes and time spent traveling between patient locations. Letting  $C_j$  denote district cost as estimated by either *TSP* or *SEP*, daily workload is estimated using Equation (76):

$$T_j = \lambda_j w + C_j. \quad (76)$$

Note that  $w_i$  has been replaced with  $w$ . We assume that the average visit duration is the same across all zip codes. While conversations with home health providers do suggest that visit duration may be correlated with socioeconomic status (which can be linked to a particular geographic area), there is not currently enough evidence to support the use of zip-code dependent service times.



Recall that  $b$  is the target daily workload of a nurse specified in units of time spent working, and  $\beta = bf$  is the target daily workload of a district being served by a team of  $f$  nurses. Let  $0 \leq \alpha \leq 1$  be an allowable variation in daily workload. Then, a contiguous district served by a team of  $f$  nurses is feasible if its workload is within the range specified by Equation (77):

$$(1 - \alpha)\beta \leq T_j \leq (1 + \alpha)\beta. \quad (77)$$

#### 4.4.4 Set partitioning formulation for the nurse districting problem

Let  $\mathcal{J}$  be the set of districts which are contiguous and satisfy the workload requirements specified in Section 4.4.3. Let  $y_j$  be a binary decision variable indicating whether district  $j$  is selected. Then, HHND can be modeled as the following set partitioning problem, denoted as  $P$ . Denote its linear relaxation as  $LP$ .

$$\text{Min } \sum_{j \in \mathcal{J}} C_j y_j \quad (78)$$

$$\text{s.t. } \sum_{j \in \mathcal{J}} \gamma_{ij} y_j = 1 \quad \forall i = 1, \dots, n \quad (79)$$

$$\sum_{j \in \mathcal{J}} y_j = m \quad (80)$$

$$y_j \in \{0, 1\} \quad \forall j \in \mathcal{J} \quad (81)$$

The objective function minimizes the total cost, as defined in Section 4.4.3, of all selected districts. Constraint set (79) ensures that each zip code is included in exactly one of the selected districts. Constraint (80) ensures that the appropriate number of districts are selected, and constraint set (81) ensures that all decision variables are binary.

#### 4.5 Solution approaches

We develop a solution approach for the set partitioning formulation of HHND that combines ideas from column generation and heuristic local search methods. The procedure begins with a subset of feasible districts, solves the linear relaxation of  $P$  on the restricted set of

districts, and then uses the dual variables associated with the linear relaxation solution to guide the neighborhood search for improving columns to add to the set. This procedure is described in Section 4.5.3. This approach demonstrates that some of the benefit typically associated with optimization approaches such as branch-and-price can be realized in the context of a heuristic approach. We also develop an initial clustering heuristic and local search improvement method that can be used to obtain and improve initial feasible solutions to  $P$ . The initial solution heuristic and local search procedure are described in Sections 4.5.1 and 4.5.2.

#### 4.5.1 Initial solution

The heuristic used to obtain an initial solution to  $P$  is denoted as  $BI$  and given in Algorithm 9. The associated parameters are defined in Table 9. Letting a zip code serve as the reference zip code  $u$ ,  $BI$  selects  $v$ , the currently unassigned zip code farthest from  $u$ , and begins building a district around  $v$ , adding adjacent zip codes until doing so would violate the maximum workload constraints. Each time a zip code is added to a district, the list of unassigned zip codes is updated. Each time a district becomes “full”, such that no additional zip codes can be added without violating one or more constraints, the currently unassigned zip code farthest from  $u$  is selected to begin a new district. The process continues until no unassigned zip codes remain. If at any iteration no adjacent zip code can be identified to add to the current district before the district has reached minimum workload feasibility, the district is marked as infeasible and the next zip code  $v$  is selected to begin a new district. This process is repeated  $n$  times, letting each zip code serve as the reference node exactly once. At the end of  $n$  iterations,  $n$  solutions to HHND are returned; i.e.,  $n$  sets of districts which cover each zip code exactly once.

This approach is similar to the clustering heuristic proposed in Mehrotra et al. [35], but differs primarily in its selection rule used at each iteration. The approach in [35] selects the zip code which is closest to the reference node, where distance is defined as the number of adjacency graph edges in the shortest path from the zip code to the reference node. Ties are broken between alternatives by selecting the zip code with maximum degree to the

district being created. Instead, our approach first identifies whether any of the adjacent zip codes which could feasibly be added to the current district are adjacent to any currently unassigned zip codes. If there exists a zip code adjacent to the current district which is not adjacent to at least one currently unassigned zip code, it is selected. If the zip code were not added to the current district, a separate district containing only that zip code would be created in a future iteration. If no such zip codes exist, the selection rule identifies the zip code with maximum degree to the current district, and breaks ties among alternatives by selecting the zip code with minimum degree to the list of unassigned zip codes.

**Table 9:** Parameters used in initial solution heuristic

Parameter	Description
$\mathcal{L}$	List of unassigned zip codes
$u$	Reference zip code
$d_{ij}$	Euclidean distance between centroids of zip codes $i$ and $j$
$p$	District being generated at each iteration
$a_{ip}$	Equal to 1 if zip code $i$ included in district $p$ and 0 otherwise
$deg[i]$	Number of zip codes in $\mathcal{L}$ that zip code $i$ is adjacent to
$deg_p[i]$	Number of zip codes in district $p$ that zip code $i$ is adjacent to
$T_{p+w}$	Workload of district $p$ if zip code $w$ is added

---

**Algorithm 9** Initial solution heuristic *BI*

---

```

1: for  $u = 1, \dots, n$  do
2:   Initialize  $a_{ip} = 0 \forall i, p$ ;  $\mathcal{L} = \{1, \dots, n\}$ ;  $p = 0$ 
3:    $v = \arg \max_i \{d_{iu} : i \in \mathcal{L}\}$ 
4:    $\mathcal{I} = \{i \in \mathcal{N} \mid (i \in \mathcal{L}) \&\& (deg_p[i] > 0) \&\& (T_{p+i} \leq (1 + \alpha)b_2)\}$ 
5:   if  $|\mathcal{I}| > 0$  then
6:     if  $deg[i] > 0$  for any  $i \in \mathcal{I}$  then
7:        $w = i$ 
8:     else
9:        $w = \arg \min_{i \in \mathcal{L}} \{deg[i] \mid deg_p[i] = \max_{i \in \mathcal{L}} deg_p[i]\}$ 
10:    end if
11:     $a_{wp} = 1$ ;  $\mathcal{L} = \mathcal{L} \setminus \{w\}$ ;  $deg_p[i]$  and  $deg[i]$  updated  $\forall i \in \mathcal{L}$ . Return to Step 4 if  $\mathcal{L}$  is not empty. Else stop.
12:  else
13:    Flag  $p$  as infeasible if  $T_p < (1 - \alpha)b_2$  by setting  $C_p = \infty$ . Set  $p = p + 1$  and return to Step 3 if  $\mathcal{L}$  is not empty. Else stop.
14:  end if
15:  Save solution associated with reference node  $u$ 
16: end for

```

---

Note that some solutions returned by Algorithm 9 may contain districts which are infeasible with respect to the minimum workload bound, and may contain more than the

desired number of districts,  $m$ . Only one solution will be passed to the local search or column generation phase. If at least one solution is available which contains exactly  $m$  districts that are feasible with respect to workload bounds, the solution with lowest total cost is selected. If no feasible solutions are available, the one with least infeasibility is selected.

A solution is said to have least infeasibility if the amount of workload, measured in time required to serve district demand, that needs to be shifted out of the infeasible districts is least. Consider what is required to convert a solution containing infeasible districts to feasibility. The zip codes in the infeasible districts must be shifted to adjacent districts, increasing the daily workload of those adjacent districts without violating the maximum workload bounds. Solutions with the least amount of workload that needs to be shifted should be among the least difficult to convert to feasibility.

The local search procedure described next contains methods for removing infeasibility.

#### **4.5.2 Local search**

Using the set of districts provided by the initial solution heuristic, local search is used to first attain feasibility if necessary, then improve the starting solution. We first explain the local search procedure for removing infeasibility, denoted as *LS1*.

##### ***LS1*: Local search procedure for removing infeasibility**

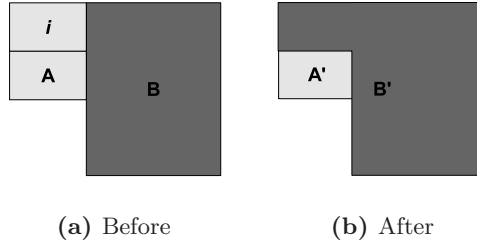
*LS1* uses two variations of a shift move used to convert an infeasible solution containing more than  $m$  districts, some of which violate minimum workload bounds, to a solution containing exactly  $m$  feasible districts:

**Type I shift** Move a zip code from a district which is infeasible with respect to minimum workload bounds to a feasible district, maintaining feasibility of the new district

**Type II shift** Move a zip code from a district which is feasible with respect to workload bounds to another feasible district, maintaining feasibility of both districts

Let  $A$  be an infeasible district which contains zip code  $i$ , and let  $B$  be a feasible district. Note that districts are simply sets of zip codes. Consider performing a type I shift to create

district  $B'$  by adding zip code  $i$  to  $B$ . The move is illustrated in Figure 13a. In a type I shift, the goal is to eliminate the infeasible district. Thus, evaluating the feasibility of the move only requires evaluating the feasibility of the new district  $B'$ . If zip code  $i$  is adjacent to at least one zip code in  $B$ ,  $B'$  will remain contiguous. It remains only to check whether  $B'$  will violate maximum workload bounds. The daily demand and area of  $B'$  are calculated by adding the demand and area of zip code  $i$  to  $B$ :  $\lambda_{B'} = \lambda_B + \rho_i$ ,  $V_{B'} = V_B + v_i$ . Then,  $\lambda_{B'}$  and  $V_{B'}$  can be used to calculate district workload as specified in Equation (76).

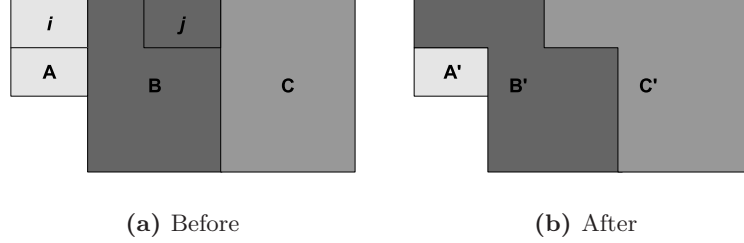


**Figure 13:** Type I shift example

In some instances, it is not possible to attain feasibility using only type I shifts. In the example shown in Figure 13, if the current workload of  $B$  were too high to accept  $i$ , and no other districts adjacent to  $A$  could accept  $i$ , it would not be possible to eliminate  $A$ . Two districts  $A$  and  $B$  are said to be **adjacent** if at least one pair of zip codes  $(i, j)$  can be found for which  $i \in A$ ,  $j \in B$ , and  $i$  and  $j$  are adjacent. Type II shift moves are designed to shift workload out of districts which are adjacent to infeasible districts, to free capacity for accepting workload from the infeasible district.

Type II shifts involve two districts which must both remain feasible after the move is performed. Let  $B$  be a feasible district containing zip code  $j$ , and let  $C$  be a feasible district adjacent to  $B$ . If districts  $B'$  and  $C'$  are created by moving  $j$  to  $C$ , district  $C'$  will remain contiguous if  $j$  was adjacent to at least one zip code in  $C$ . However,  $B'$  must be checked for contiguity, because removing  $j$  could possibly disconnect other zip codes in the district. To check whether  $B'$  and  $C'$  violate workload bounds, we must ensure that removing  $j$  from  $B$  does not cause  $B'$  to violate minimum workload bounds, and adding  $j$  to  $C$  does not cause  $C'$  to violate maximum workload bounds. The workload of  $B'$  and  $C'$  can be calculated

using Equation (76) with the following inputs:  $\lambda_{B'} = \lambda_B - \rho_j$ ,  $\lambda_{C'} = \lambda_C + \rho_j$ ,  $V_{B'} = V_B - v_j$ ,  $V_{C'} = V_C + v_j$ . Figure 14 illustrates a type II shift between districts  $B$  and  $C$  that enables a type I shift between districts  $A$  and  $B'$ .



**Figure 14:** Type II shift followed by type I shift example

The local search procedure for removing infeasibility explores type I and type II shift neighborhoods as follows. Let:

- $\mathcal{I}$  contain all infeasible districts,
- $\mathcal{J}$  contain all feasible districts adjacent to some district in  $\mathcal{I}$ , and
- $\mathcal{K}$  contain all feasible districts adjacent to some district in  $\mathcal{J}$ .

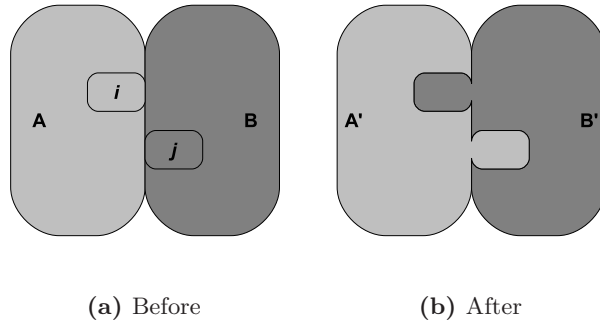
First, perform all possible type I shifts between pairs of districts in  $\mathcal{I}$  and  $\mathcal{J}$ , updating each set between each move. When no additional type I shifts are possible, if  $\mathcal{I}$  is not empty, perform type II shifts between pairs of districts in  $\mathcal{J}$  and  $\mathcal{K}$ , updating each set between each move. When no additional type II shifts are possible, perform type I shifts again between pairs of districts in  $\mathcal{I}$  and  $\mathcal{J}$ . Iterate in this manner until  $\mathcal{I}$  is empty or no additional type I or type II shifts are possible.

During the execution of *LS1*, the district cost associated with each move is not evaluated because achieving feasibility is the primary concern. This procedure is only used when the solution provided by *BI* is not feasible. If *LS1* is also unable to provide a feasible solution, the parameter  $\alpha$  is increased to allow a wider variation in workload bounds. If the solution provided by *LS1* is feasible, it is passed to the following local search and column generation procedures.

**LS: Local search procedure for improving a feasible solution**

We develop a local search procedure denoted as *LS* to improve an existing feasible solution; i.e., a set of  $m$  feasible districts provided by either *BI* or *LS1*. Two neighborhoods are considered in the procedure. The first neighborhood includes all feasible solutions which can be reached from the current solution by performing a single **shift**, equivalent to the type II shift presented in the previous section. The only difference between a type II shift in *LS1* and a shift in *LS* is that in order for a move to be implemented in *LS*, it must be improving in addition to being feasible. A move is improving if the cost of the district created by the move is strictly less than the cost of the original district. Letting  $A'$  denote the new district created from  $A$  by performing a shift,  $C_{A'}$  must be strictly less than  $C_A$ .

The second neighborhood includes all feasible solutions which can be reached by performing a single **swap**: exchanging a pair of zip codes between a pair of adjacent districts. Figure 15 illustrates a swap. Zip code pair  $i$  and  $j$  is swapped between district pair  $A$  and  $B$  to create new districts  $A'$  and  $B'$ .



**Figure 15:** Swap example

In order for a swap to be feasible, both districts created by the swap must be feasible. Denote the district created by removing  $i$  from  $A$  and adding  $j$  as  $A'$ , and denote the district created by removing  $j$  from  $B$  and adding  $i$  as  $B'$ . Districts  $A'$  and  $B'$  must be checked for contiguity because removing zip codes  $i$  and  $j$  respectively could disconnect other zip codes in the districts. The workload of  $A'$  and  $B'$  can be calculated using the following inputs:  $\lambda_{A'} = \lambda_A - \rho_i + \rho_j$ ,  $\lambda_{B'} = \lambda_B + \rho_i - \rho_j$ ,  $V_{A'} = V_A - v_i + v_j$ ,  $V_{B'} = V_B + v_i - v_j$ . Both the maximum workload and minimum workload feasibility of both districts must be ensured, because it is not certain whether the net change in workload will be positive or negative.

In order for a feasible swap to be implemented, the total cost of districts  $A'$  and  $B'$  must be strictly less than the total cost of districts  $A$  and  $B$ ;  $C_{A'} + C_{B'} < C_A + C_B$ .

The local search procedure for improving a feasible solution via the shift and swap neighborhoods is as follows. Assume that a feasible set of  $m$  districts is passed to  $LS$  from  $BI$  or  $LS1$ . Let  $shift()$  be a subroutine which explores all district pairs for improving shifts, implementing the most improving shift if multiple alternatives are found and updating the current set of districts to reflect the change. Let the subroutine return a value of 1 if an improving shift is found and 0 otherwise. Let  $swap()$  be the complementary swap subroutine. Then, the local search method is detailed in Algorithm 10.

---

**Algorithm 10** *LS*: Local search procedure

---

```

1: Initialize  $swapCount = 1$ ,  $shiftCount = 1$ ,  $y = 1$ ,  $z = 1$ 
2: while  $shiftCount + swapCount > 0$  do
3:    $shiftCount = 0$ ,  $swapCount = 0$ ;
4:   while  $z \neq 0$  do
5:      $z = swap()$ ;
6:      $swapCount = swapCount + z$ ;
7:   end while
8:   while  $y \neq 0$  do
9:      $y = shift()$ ;
10:     $shiftCount = shiftCount + y$ ;
11:   end while
12: end while

```

---

### 4.5.3 Column generation procedure

Consider problem  $P$ , the integer programming problem given in Section 4.4.4, which includes a binary decision variable for each feasible district. The number of feasible districts is exponential in the number of zip codes. Thus, instead of solving the IP on the full set  $\mathcal{J}$ , we propose a column generation approach, where feasible districts are generated and added to the model as needed.

The column generation approach begins with an initial set of feasible districts  $\mathcal{J}'$  obtained using either  $BI$ ,  $LS1$ , or  $LS$ , and solves the linear relaxation of  $P$  on the restricted set of columns. We refer to the column generation procedure as  $CG + BI$ ,  $CG + LS1$ , and  $CG + LS$ , depending on how the set  $\mathcal{J}'$  was obtained:



- When  $BI$  provides a feasible solution, the set  $\mathcal{J}'$  contains the districts which comprise the corresponding solution, and we refer to the column generation procedure as  $CG + BI$ .
- When  $BI$  does not provide a feasible solution but  $LS1$  is able to modify the solution provided by  $BI$  to attain feasibility,  $\mathcal{J}'$  contains the set of districts provided by  $LS1$  and we refer to the column generation procedure as  $CG + LS1$ .
- When neither  $BI$  nor  $LS1$  provide a feasible solution for an instance of HHND and parameter  $\alpha$ , the column generation procedure is not used. While passing a solution which contains infeasible districts which have been penalized with high cost may allow the column generation procedure to achieve a feasible solution, experiments indicated the approach was not effective.
- Each time a feasible solution is provided by  $BI$  or  $LS1$ ,  $LS$  is used to improve the solution. Then, the column generation procedure is used to improve the best solution provided by  $LS$ . In this case,  $\mathcal{J}'$  contains the districts corresponding to the best solution provided by  $LS$  and we refer to the column generation procedure as  $CG + LS$ .

The linear relaxation of  $P$  on the restricted set of columns  $\mathcal{J}'$  is denoted as  $LPR$ , given in Equations (82) - (85).

$$\text{Minimize } \sum_{j \in \mathcal{J}'} C_j y_j \quad (82)$$

$$\text{Subject to } \sum_{j \in \mathcal{J}'} \gamma_{ij} y_j = 1 \quad \forall i = 1, \dots, n \quad (83)$$

$$\sum_{j \in \mathcal{J}'} y_j = k \quad (84)$$

$$0 \leq y_j \leq 1 \quad \forall j \in \mathcal{J}' \quad (85)$$

An optimal solution  $\bar{y}$  to  $LPR$  provides a set of dual values  $\pi_i$  for  $i = 1, \dots, n + 1$ . The first  $n$  dual values correspond to the zip code covering constraints given in Equation (83), and the last corresponds to the cardinality constraint given in Equation (84). This solution

is only optimal to  $LP$  on the full set of columns  $\mathcal{J}$  if no columns  $j \in \mathcal{J} \setminus \mathcal{J}'$  have negative reduced cost. The reduced cost of a column  $\bar{C}_j$  is given by:

$$\bar{C}_j = C_j - \sum_i \pi_i \gamma_{ij} - \pi_{n+1}. \quad (86)$$

Testing for the optimality of  $\bar{y}$  to the linear relaxation of  $P$  on the full set of districts,  $\mathcal{J}$ , requires solving the pricing problem given in Equation (87):

$$\min \left\{ \bar{C}_j - \sum_{i \in \mathcal{N}} \pi_i \gamma_{ij} - \pi_{n+1} \mid j \in \mathcal{J} \right\}. \quad (87)$$

If any districts  $j$  are found such that  $\bar{C}_j < 0$ , these districts represent columns which could enter the basis and improve the LP solution.

Recall that feasible districts must satisfy workload bounds and contiguity constraints. Thus, solving the pricing problem requires solving a two-sided knapsack problem with side constraints. The pricing problem encountered in Mehrotra et al. [35] has a similar form. The authors decomposed the pricing problem into  $i = 1, \dots, n$  subproblems, one for each subunit, where the solution to each subproblem  $i$  provided the minimum cost district rooted at subunit  $i$ . Due to the linear form of the knapsack constraints and cost functions used in their problems, they were able to solve the subproblems corresponding to each subunit optimally until a solution yielded a negative reduced cost column. However, in our problem, both the knapsack constraints and the cost function in the pricing problem are nonlinear. Instead of solving the pricing problem optimally, we search for columns to add heuristically, using the dual values associated with the current solution to  $LPR$  to guide the neighborhood exploration. Once a number of improving moves of each type have been implemented or a number of potential moves of each type have been considered during the current execution of neighborhood search for the pricing problem,  $LPR$  is re-solved on the augmented set  $\mathcal{J}'$ . This process continues until all dual variables are nonnegative, or until a number of neighborhood search executions have been performed. Let  $EN$  denote the limit on the number of neighborhood search executions.

The outline of the column generation approach is given in Algorithm 11. The heuristic

method we use to search for new columns to add at each iteration is described below.

---

**Algorithm 11** Column generation procedure

---

```

1: Let  $k = 0$ . Solve  $LPR$  on  $\mathcal{J}'$  to get  $\bar{y}$  and  $\bar{\pi}$ 
2: if  $k < EN$  and  $\pi_i < 0$  for any  $i \in \mathcal{N}$  then
3:   use neighborhood search to find feasible districts for which  $\bar{C}_j < 0$ ; let  $k = k + 1$ 
4:   if step 3 identified improving districts then
5:     add those districts to  $\mathcal{J}'$  and return to step 1
6:   else
7:     go to step 10
8:   end if
9: end if
10: if the current solution  $\bar{y}$  to  $LPR$  is integral then
11:   stop.
12: else
13:   use branch and bound to solve  $P$  using column set  $\mathcal{J}'$ 
14: end if

```

---

### Solving the pricing problem with neighborhood search

A variety of local search moves are included in the neighborhood search procedure used to solve the pricing subproblem. There is some similarity between the moves used in the column generation neighborhood search and those used in local search. However, key differences affect both the types of moves used in  $CG$  and the manner in which neighborhoods are explored:

- A single integer-feasible solution to the set partitioning problem  $P$  is maintained at each iteration of  $LS$ . Thus, the number of feasible districts on hand at any iteration is  $m$ , and each zip code is included in exactly one district.
- When two new districts denoted as  $A'$  and  $B'$  are created by modifying two existing districts  $A$  and  $B$  using  $LS$ , the original districts are deleted. Because an integer feasible solution is always maintained, this implies that  $A \cup B = A' \cup B'$  for all feasible moves.

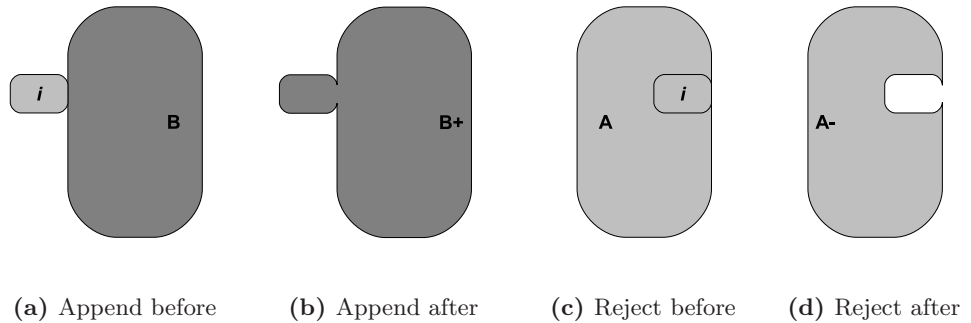
In the column generation neighborhood search procedure, districts are never deleted from the set  $\mathcal{J}'$ , therefore the number of columns on hand can be very large. Also, an integer-feasible solution to  $P$  is not necessarily available during column generation. If the solution  $\bar{y}$  to  $LPR$  is not integer-feasible at some iteration, there must exist at least one

zip code which is included in multiple districts which each have fractional value  $y_j$  and zero reduced cost. A move such as “shift zip code  $i$  from its current district to an adjacent district” can require evaluating numerous shift alternatives when multiple districts with  $y_j > 0$  contain  $i$ . Thus, it is important to design a careful neighborhood exploration method, and the method should focus on moves which are likely to generate improving columns.

Furthermore, care must be taken to ensure that at least some of the improving columns being added to  $\mathcal{J}'$  could be feasibly selected along with a set of complementary columns, possibly during the next solution of the linear programming master problem  $LPR$ , to create an integer-feasible solution to  $P$ . Thus, we explore a swap neighborhood which creates pairs of districts that are complementary to an existing pair of districts. Other moves are intended to diversify the column pool. The neighborhoods that we use to generate moves, in addition to the swap neighborhood, are: append, reject, and append/reject. In each case, the districts created from the move must be contiguous, feasible with respect to workload bounds, and have negative reduced cost. Feasibility conditions are discussed in the context of each type of neighborhood.

#### Append neighborhood and reject neighborhood

The **append** neighborhood includes all districts which can be created by adding a single zip code to an existing district. The **reject** neighborhood includes all districts which can be created by removing a single zip code from an existing district. The moves are illustrated in Figure 16.



**Figure 16:** Append example and reject example

Let  $B'$  be a district that is created by appending  $i$  to feasible district  $B$ . District  $B'$  will be contiguous if zip code  $i$  is adjacent to at least one zip code in  $B$ . The cost and workload feasibility of  $B'$  can be calculated using the following inputs:  $\lambda_{B'} = \lambda_B + \lambda \rho_i$ ,  $V_{B'} = V_B + v_i$ . The move is feasible if district  $B'$  is contiguous and does not violate maximum workload bounds. The append will be implemented if  $B'$  is feasible and  $\bar{C}_{B'}$  is negative.

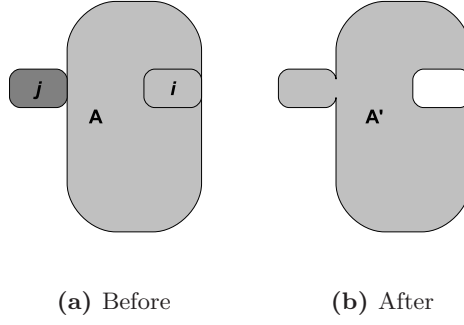
Let  $A'$  be a district that is created by rejecting  $i$  from feasible district  $A$ . District  $A'$  must be checked for contiguity because removing  $i$  may disconnect some zip codes in the district. The cost and workload feasibility of  $A'$  can be calculated using the following inputs:  $\lambda_{A'} = \lambda_A - \rho_i$ ,  $V_{A'} = V_A - v_i$ . The move is feasible if district  $A'$  is contiguous and does not violate minimum workload bounds. The reject will be implemented if  $A'$  is feasible and  $\bar{C}_{A'}$  is negative.

The append and reject neighborhoods are explored in the following manner. First identify a district  $J$  with zero reduced cost. If any zip code  $i$  which is adjacent to a zip code in  $J$  but not included in  $J$  has positive dual value, evaluate the cost and feasibility of creating a new district by appending  $i$  to  $J$ . Add all improving columns to  $\mathcal{J}'$ . For any zip code  $k$  which has positive dual value and is currently included in  $J$ , evaluate the cost and feasibility of creating a new district by rejecting  $k$  from  $J$ . Add all improving columns to  $\mathcal{J}'$ . Iterate through some number of zero reduced cost districts selected at random, continuing the search for append and reject moves until some number of improving moves have been implemented.

### **Append/reject neighborhood**

The append/reject neighborhood explores moves which combine an append move and reject move into a single operation. Given a district  $A$  which currently includes zip code  $i$  but not  $j$ , zip code  $i$  is rejected and zip code  $j$  is appended. The move is illustrated in Figure 17.

District  $A'$  created by an append/reject move is feasible if it is contiguous, and if it does not violate minimum and maximum workload bounds. The net change in district workload may be positive or negative. The cost and workload feasibility of  $A'$  can be calculated using the following inputs:  $\lambda_{A'} = \lambda_A + \rho_j - \rho_i$ ,  $V_{A'} = V_A - v_i + v_j$ . The append/reject will be



**Figure 17:** Append/reject example

implemented if  $A'$  is feasible and  $\bar{C}_{A'}$  is negative.

The append/reject neighborhood is only explored for a given district when no append moves are feasible for that district. Thus, it is used as a subroutine within the append neighborhood exploration procedure.

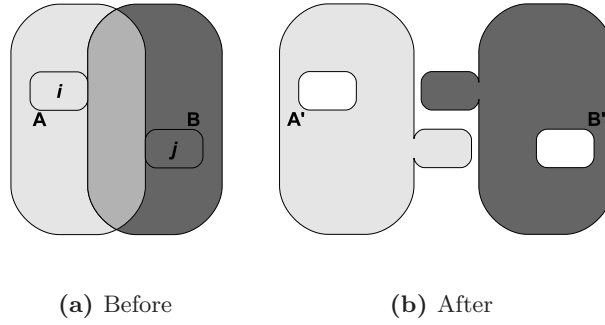
### Swap neighborhood

The swap neighborhood generates columns which are complementary to a pair of columns currently in  $\mathcal{J}'$ , and is the only local search move in the column generation procedure that requires identifying a pair of districts which are adjacent and disparate. It is similar to the swap defined within the context of local search, differing only in the manner in which the neighborhood is explored.

In our search for swaps, we can clearly restrict the search to those pairs of districts which are adjacent. Otherwise, the resulting districts can not be contiguous. In the local search procedure, any two districts in the current solution were guaranteed to be disparate, because the current solution was integer-feasible to the set partitioning problem  $P$ . In the set  $\mathcal{J}'$  in the column generation procedure, two adjacent districts are not guaranteed to be disparate, because each zip code may appear in multiple districts. Performing a swap between two districts which are adjacent but not disparate is undesirable.

First consider the example shown in Figure 18, where the intersection between districts  $A$  and  $B$  is shaded in medium gray. Assume the swap is feasible with respect to workload bounds. The newly created districts  $A'$  and  $B'$  remain contiguous, therefore they are feasible. However, recall that in any feasible solution to problem  $P$ , which is an IP, every

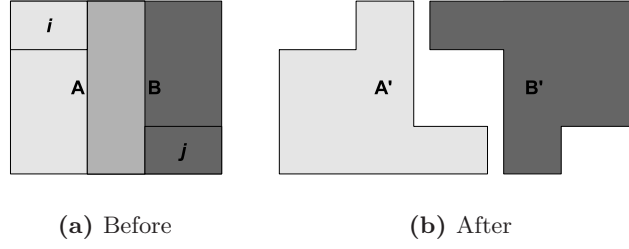
zip code must be included in exactly one district. Thus, the complement of the zip codes covered by each district selected in the solution to  $P$  must be covered by other selected districts. Neither  $A'$  nor  $B'$  can be selected in a solution to  $P$  because it is not possible to simultaneously select a contiguous district serving  $i$  or  $j$  that does not also include some of the zip codes in  $A'$  or  $B'$ . Adding  $A'$  and  $B'$  to the set  $\mathcal{J}'$  would add to the computational burden of the column generation procedure without any promise of being useful.



**Figure 18:** Infeasible swap example

It is possible that excluding swaps between pairs of non-disparate districts can cut off portions of the feasible region. However, districts that can be created by performing a swap between two non-disparate districts can also be created by a sequence of append/reject moves. Consider the example shown in Figure 19, where the intersection between  $A$  and  $B$  is shown in medium gray. Assume that districts  $A'$  and  $B'$ , which result from swapping zip codes  $i$  and  $j$ , are feasible with respect to workload bounds.  $A'$  could also have been obtained by performing an append/reject on  $A$ , appending  $j$  and rejecting  $i$ .  $B'$  could have been obtained by performing an append/reject on  $B$ , appending  $i$  and rejecting  $j$ . The append/reject moves are feasible with respect to workload bounds if the corresponding swap is feasible with respect to workload bounds. Note that at most one of the districts  $A'$  and  $B'$  can be selected in a solution to  $P$ .

The swap neighborhood is explored in the following manner. First identify a district which has zero reduced cost and contains some district with positive dual value. Identify a second district which has zero reduced cost and is adjacent and disparate to the first. Because identifying a pair of such districts is computationally expensive but searching for



**Figure 19:** Swap that could be created by append/reject

swaps between the pair is not, perform all feasible improving swaps between pairs of zip codes in the pair of districts. Iterate randomly through some number of districts, until an upper limit on the number of swaps implemented in a single execution of neighborhood search is reached.

### Implementation issues

Various parameters are used in our column generation implementation. First, upper limits are placed on both the number of potential moves of each type considered and the number of implemented moves of each type which result in improving columns being added during a single execution of neighborhood search for solving the pricing subproblem. The limits reduce neighborhood exploration time, and are also likely to reduce the time required to solve the final IP by preventing unnecessary columns from being added to  $\mathcal{J}'$ . There is some benefit associated with solving *LPR* more frequently, because it provides updated information that can be used to guide neighborhood search. Our experiments demonstrated that limits of 1000 on the number of potential moves of each type considered and 200 on the number of moves of each type implemented were effective in balancing computation time with the quality of solutions obtained. We also place an upper limit  $EN$  on the total number of executions of the neighborhood search procedure.

When the column generation procedure begins, it is possible alternatively to explore the local search neighborhoods exhaustively because the set  $\mathcal{J}'$  is very small. Thus, we permit the heuristic to use exhaustive search during the first execution of neighborhood search in some experiments. We refer to this as a “warmup” and denote the warmup parameter as  $r$ , where  $r = 0$  indicates the first execution of neighborhood search is identical to all others,



and  $r = 1$  indicates neighborhoods are explored exhaustively during the first execution. Section 4.6 presents the results of a computational study.

#### 4.6 Experiments and results

We perform a computational study on a set of test problems to compare the quality of nurse districts developed using local search and column generation heuristics. We also perform a simulation study to compare district cost estimation methods *TSP* and *SEP* with the cost of simulated hamiltonian paths through actual patient locations in districts which comprise heuristic solutions. Problem instances are based on data provided by a home health agency that employs 160 nurses and covers a 5,500 square mile service region comprised of 156 zip codes. A map displaying boundaries of zip codes included in the service region is used to determine zip code adjacencies. Zip codes within boundaries of the service region but not currently served by the home health agency are included in the adjacency graph to ensure the adjacency subgraph corresponding to each district created by the solution procedures is connected. Such zip codes are assigned a daily demand of 0.

A snapshot of patients receiving service from the home health agency in July 2008 was provided. A total of 1415 patients were enrolled. Table 10 summarizes characteristics of zip codes included in the service region. The minimum and maximum area, patient count, and patient density across all zip codes is reported. Also reported is the sample mean and standard deviation for each metric.

**Table 10:** Characteristics of zip codes in test instance service region

	Min	Max	Sample mean	Std dev
Area (mi <sup>2</sup> )	0.20	244	35.6	38.4
Patient count	1	48	9.12	8.64
Patient density (pts/mi <sup>2</sup> )	0.004	5.613	0.854	1.310

Nominal daily demand estimates are developed for each zip code using the provided patient enrollment snapshot. Given the weekly visit frequency of each patient enrolled, the average visit request rate across all patients was calculated to be 0.67 visits per patient per day. The visit request rate per patient is assumed to be homogenous throughout the service region. This is a reasonable assumption, as patient diagnoses are the primary indicators

of patient visit frequencies, and patient diagnoses patterns should not vary dramatically across the region being served by a single agency. We also assume that patient visits have the same duration in each zip code,  $w_i = w = 30$  minutes.

Letting  $p_i$  denote the current patient count in each zip code, the number of visits required to each zip code each day is  $\rho_i = 0.67p_i$ . Districts are planned assuming the nominal demand level in each zip code is static. This mimics district planning processes currently employed by home health agencies; districts are developed for a current realization of demand, and are re-planned when the result of intermittent workload reviews indicate disparity in nurse workload. An alternate nominal demand estimate that may provide more robust district plans would adjust the current demand according to a projection of future demand based on historical data, but no such information was available for the purposes of this study.

Recall that nurses work in teams of size  $f$  to cover demand within the district to which they have been assigned. Workload targets are calculated as a function of the instance being solved and are assumed to be reasonable in all problem instances studied, *i.e.*, standard workday lengths result. In our study, district workloads are required to be within a 10% deviation from the district workload target ( $\alpha = 0.1$ ). Experiments revealed that smaller bounds for  $\alpha$  were not achievable for all problem instances studied.

To determine target workloads for each district, we assume that total demand in the service area can be equally divided among  $f$  nurses. This yields an estimate for the total amount of time spent performing patient visits each day in each district equal to  $fw\beta_1$ , where  $\beta_1$  is calculated as:

$$\beta_1 = \frac{\sum_{i=1}^n \rho_i}{k}. \quad (88)$$

An estimate for the time spent traveling in each district, which is a component of total target workload, is developed separately for methods *TSP* and *SEP*. For method *TSP*, Equation (74) is used to estimate district target travel time when the service area and total demand are equally divided among districts. Then, total target workload for a single district is denoted  $\beta_2^{TSP}$  and is calculated using Equation (89):

$$\beta_2^{TSP} = fw\beta_1 + 0.7214f \left(1 - \frac{1}{\beta_1}\right) \sqrt{\frac{\beta_1 \sum_{i=1}^n v_i}{m}}. \quad (89)$$

For method *SEP*, the average separation between points in each district is calculated for a set of test districts known to be feasible for the given instance. Then, the average across all districts, denoted as  $h'$ , is calculated. The resulting target total workload for a single district is denoted as  $\beta_2^{SEP}$  and is calculated using Equation (90):

$$\beta_2^{SEP} = fw\beta_1 + fh'(\beta_1 - 1). \quad (90)$$

Districts are feasible with respect to workload bounds when  $T_j$  is within the allowable interval  $[0.9\beta_2, 1.1\beta_2]$ . In summary, each district must satisfy:

$$0.9\beta_2 \leq \lambda_j w + C_j \leq 1.1\beta_2. \quad (91)$$

Note that if the objective were to balance patient visit count across districts,  $\beta_1$  could be used as the district workload target, and district workload would be equal to district demand ( $T_j = \lambda_j$ ).

We use team sizes of 5 and 10 nurses in our computational study. Test instance data prevented solving HHND for teams of less than 5 nurses, because the workload of some zip codes in the service area is above maximum workload feasibility bounds. Results are presented for method *TSP* for the  $f = 5$  and  $f = 10$  instances and for method *SEP* for the  $f = 5$  instance following a discussion of heuristic parameters used, and a description of the simulation study performed to validate routing cost estimates.

#### 4.6.1 Heuristic parameters

The *CG* parameters we use in our test study include warmup values of 0 and 1, denoted as  $r$ . We use an upper limit of 200 on the number of implemented moves of each type and an upper limit of 1000 on the number of potential moves of each type explored at each execution of neighborhood search for solving the pricing subproblem. We experiment with limiting the total number of executions of neighborhood search by 10, 20, and 40 iterations,

and denote this parameter as  $EN$ . Three replications of each experiment are performed using three different random seeds, denoted as  $s = 1, 2$ , and  $3$ .

We experiment with passing the first feasible solution identified and the best feasible solution identified to  $CG$  to comprise the set  $\mathcal{J}'$ . The first feasible solution is obtained via  $BI$ , unless  $LS1$  is required, and the best feasible solution passed to  $CG$  is obtained via  $LS$ . Note that a feasible solution must also be passed to  $LS$  from  $BI$  or  $LS1$ .

- When a feasible solution is obtained using  $BI$ , both  $LS$  and  $CG$  are used to improve the solution. Additionally,  $CG$  is used to improve the best solution obtained using  $LS$ . Thus, results are reported for:  $BI$ ,  $LS$ ,  $CG + BI$ ,  $CG + LS$ .
- When  $BI$  does not return a feasible solution,  $LS1$  is used to obtain a feasible solution. Then,  $LS$  and  $CG$  are used to improve the solution from  $LS1$ , and  $CG$  is additionally used to improve the solution from  $LS$ . Thus, results are reported for:  $LS1$ ,  $LS$ ,  $CG + LS1$ ,  $CG + LS$ .

For each problem instance, the best solution achieved by each heuristic is reported, and additional details are given regarding the best overall solution obtained. For the column generation procedure, we also report the solution value and total computation time that are observed for each combination of parameters. Results are discussed separately for the 5 and 10 nurse instances.

#### 4.6.2 Comparison of district cost approximations to simulated Hamiltonian path costs

A simulation is performed for the 5-nurse instance to test the quality of routing cost approximation methods  $TSP$  and  $SEP$  using a set of actual patient addresses in the service area. For each district included in the best solutions obtained when methods  $TSP$  and  $SEP$  are used in combination with the column generation heuristic,  $C_j^{TSP}$  and  $C_j^{SEP}$  are compared, respectively, with the sum of the costs of the Hamiltonian paths through five sets of  $\frac{\lambda_j}{5}$  randomly selected patient addresses in the district. The Hamiltonian path cost for each set of addresses is obtained by subtracting the longest arc from the TSP tour returned by the Concorde TSP solver [18]. 100 random replications are performed for each district.

#### 4.6.3 Results for 5-nurse instance, $TSP$

We first present results regarding the best solutions achieved for the 5-nurse instance, followed by simulation study results and performance information for the column generation procedure.

##### Solution quality

In this set of experiments, 32 districts are designed for teams of  $f = 5$  nurses. Workload bounds are set at [920,1140] minutes. Table 11 reports the best solutions obtained by  $LS1$ ,  $LS$ ,  $CG + LS1$ , and  $CG + LS$ . For the  $CG$  heuristics, the value reported corresponds a single observation from the random seed which returned the best solution. Note that the  $LS$  heuristic values reported correspond to a single observation as well, because those heuristics do not use random seeds to perform neighborhood search. The initial feasible solution provided by  $LS1$  has total cost 4931. The best solution obtained for this instance with total cost 4723 is provided by  $CG + LS1$ . This solution was shown to be locally optimal by passing the final solution to  $LS$  and verifying that no additional improvements were identified.

**Table 11:** Best solution obtained using each heuristic,  $TSP$ ,  $f = 5$

	Sol Val	% imp. over $LS1$	% imp. over $LS$
$LS1$	4931	-	-
$LS$	4871	1.2%	-
$CG + LS1$	4725	4.2%	3.0%
$CG + LS$	4723	4.2%	3.0%

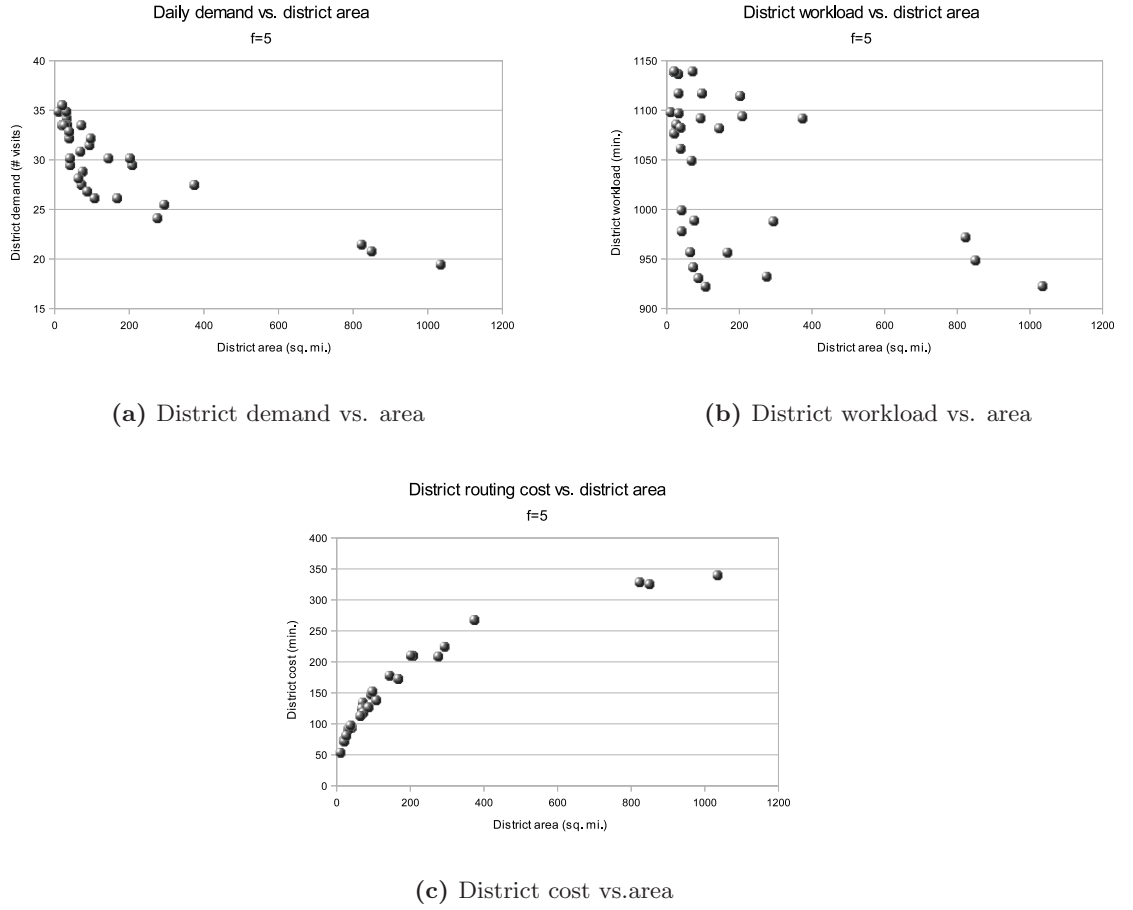
The parameters for  $CG + LS1$  which were used to obtain the best solution were  $r = 0$ ,  $s = 1$ , and  $EN = 20$ . Table 17 gives the number of moves of each type which were implemented in the corresponding experiment, and the number of districts corresponding to each type of move which were selected in the final solution to  $P$ . Appends, rejects, append/rejects, and swaps are denoted by A, R, AR, and S, respectively. The initial districts  $I$  selected in the final solution are included for completeness, so that the numbers in the corresponding row add to 32. The type of move both implemented most frequently, and having associated districts selected most frequently in the final solution is append/reject.

For the best solution obtained for this instance, Figure 20 displays district daily demand,

**Table 12:** Moves of each type implemented and selected,  $TSP$ ,  $f = 5$

	A	R	A/R	S	I	Total
# implemented	417	159	2500	428	-	3504
# selected in final solution	4	4	13	2	9	32

district daily workload, and routing cost as a function of district size. As expected, routing costs increase and the number of daily visits decrease as the district size increases. In Figure 20b, note that the y-axis has been adjusted because workload bounds constrain district workload to be within  $[920, 1140]$ . The largest districts tend to have workloads nearer to the lower bound on allowable workload.



**Figure 20:** District demand, workload, and cost vs. district area,  $TSP$ ,  $f = 5$

The district maps associated with the best solutions achieved by  $LS1$ ,  $LS$ , and  $CG + LS$  are given in Figures 24 - 26 at the end of this chapter, where each district is shaded by color.

A limited number of shades were used to enhance the appearance of the maps. Groups of zip codes which share the same color but are not contiguous comprise different districts. In each map, the gray area is not part of the home health agency’s service region.

The districts provided by *LS1* tend to have the most “regular” shape; this is expected because the initial solution heuristic builds districts by selecting zip codes to add which have the maximum degree with zip codes already included in the district. One potential drawback of *LS* and *CG* when method *TSP* is used to estimate district cost is that the districts created appear to be more irregular.

### Routing cost approximation quality

Results from the simulation study summarized in Table 13 reveal that simulated routing costs through actual patient addresses in the “irregular” districts created by *CG+LS* are higher than the simulated costs in districts created by *BI*. The “improved” solution, with respect to the *TSP* approximation, is not improved with respect to simulated travel costs. Thus, the *TSP* approximation does not do a good job of tracking simulated travel costs for this instance.

**Table 13:** Comparison of simulated hamiltonian path costs with *TSP* approximation for heuristic solutions,  $f = 5$

Heuristic	Simulated cost	<i>TSP</i> approx.
<i>BI</i>	4931	5133
<i>LS</i>	4871	5349
<i>CG+LS</i>	4723	5270

Table 14 presents the cost and workload of each district according to both the simulation study and the *TSP* approximation. In the table, the second and third columns give the area of each district and the number of patient addresses included. The fourth and fifth columns give the *TSP* district cost and workload approximations, and tables six through eight give the simulated costs and workloads. Columns nine and ten report the absolute and percent difference between the approximated and simulated workloads. Note that with workload bounds of [920,1140], the districts do not remain feasible with respect to workload bounds when simulated costs and workloads are used. For example, district 11 has a simulated workload of 835 minutes. The mean absolute difference between simulated

and approximated workload is 27.7, and the mean absolute percent difference is 2.7%.

**Table 14:** Comparison of simulated district costs and workloads with *TSP* approximation for *CG+LS* solution,  $f = 5$

$j$	District		<i>SEP</i> approx.		Simulated			Workload diff.	
	Area	Addresses	Cost	Work	Cost ( $\mu$ )	Cost ( $\sigma$ )	Work	Abs.	%
1	93.5	20	147.2	1091.9	133.8	15.2	1078.5	13.5	-1.2
2	107.2	69	138.1	922.0	160.3	16.2	944.2	22.2	2.3
3	207.7	54	209.6	1094.0	242.1	22.0	1126.5	32.5	2.9
4	71.4	83	134.2	1139.2	156.3	15.2	1161.3	22.1	1.9
5	75.6	70	124.4	988.7	177.0	19.2	1041.3	52.6	5.0
6	19.7	15	73.3	1138.6	86.6	8.3	1151.9	13.3	1.2
7	10.6	28	53.0	1098.2	54.3	6.4	1099.5	1.3	0.1
8	68.7	74	124.4	1049.0	153.7	16.2	1078.3	29.3	2.7
9	38.8	39	96.2	1061.0	97.5	11.9	1062.3	1.3	0.1
10	72.4	34	117.7	941.8	136.3	15.0	960.4	18.6	1.9
11	849.9	49	325.5	948.6	211.9	27.5	835.0	113.5	-13.6
12	1034.9	96	339.8	922.7	422.4	57.4	1005.3	82.6	8.2
13	144.0	59	177.3	1081.8	155.2	17.9	1059.7	22.2	-2.1
14	374.3	56	267.6	1091.7	249.7	25.9	1073.8	17.9	-1.7
15	32.6	53	91.9	1117.0	107.8	14.6	1132.9	15.9	1.4
16	293.7	39	224.2	988.0	258.1	30.5	1021.9	33.9	3.3
17	275.5	119	208.5	932.1	243.1	32.2	966.7	34.6	3.6
18	31.4	62	91.4	1136.6	103.7	12.9	1148.9	12.3	1.1
19	64.2	76	112.7	956.9	150.5	15.0	994.7	37.7	3.8
20	41.5	30	93.6	978.0	99.1	9.1	983.5	5.5	0.6
21	20.8	56	72.4	1077.4	91.9	13.1	1096.9	19.5	1.8
22	20.1	56	74.0	1139.3	105.9	9.0	1171.2	31.9	2.7
23	167.1	82	172.4	956.3	189.7	20.0	973.6	17.3	1.8
24	97.0	75	152.2	1117.0	148.9	16.9	1113.7	3.3	-0.3
25	41.0	20	94.6	999.1	113.3	19.5	1017.8	18.6	1.8
26	20.2	79	71.4	1076.4	95.8	9.8	1100.8	24.4	2.2
27	33.5	76	92.0	1097.0	114.6	10.9	1119.6	22.6	2.0
28	201.9	87	209.9	1114.4	269.3	31.3	1173.8	59.3	5.1
29	87.1	61	126.8	930.8	150.3	18.3	954.3	23.5	2.5
30	25.8	52	80.6	1085.6	110.8	13.1	1115.8	30.2	2.7
31	38.6	42	97.3	1082.2	104.6	10.0	1089.5	7.2	0.7
32	823.0	93	328.6	971.8	376.0	67.6	1019.2	47.4	4.6
			4723.1		5270.1				

### Column generation procedure performance

Table 15 displays the solution value observed and the total runtime in seconds of the column generation procedure for each combination of parameters. For a fixed  $r$  and  $s$ , solution values are non-decreasing as  $EN$  increases, because the same columns are identified during the first 10 pricing problem executions when  $EN = 20$  as when  $EN = 10$ , for example. Clearly, runtime increases as  $EN$  increases. The best solution achieved in 10 executions of the pricing problem is provided by  $CG + LS1$  with no warmup. The best solution achieved in 20 and 40 executions are provided by  $CG + LS$  with no warmup.

Figure 21 displays the pricing problem execution time in seconds per execution for  $CG + LS1$  and  $CG + LS$ , respectively. Observed values for each combination of runtime



**Table 15:** Column generation solution values and runtimes,  $TSP$ ,  $f = 5$ 

Method	$r$	$EN$	Solution value			Runtime (sec.)		
			$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$CG + LS1$	0	10	4769	4774	4780	16	23	9
$CG + LS1$	0	20	4765	4730	4750	75	122	62
$CG + LS1$	0	40	4748	4725	4748	507	402	250
$CG + LS1$	1	10	4788	4790	4772	17	14	22
$CG + LS1$	1	20	4749	4740	4730	85	99	122
$CG + LS1$	1	40	4749	4729	4730	288	416	395
$CG + LS$	0	10	4772	4802	4773	12	19	13
$CG + LS$	0	20	4723	4730	4739	111	90	87
$CG + LS$	0	40	4723	4729	4727	332	384	338
$CG + LS$	1	10	4827	4827	4817	5	4	8
$CG + LS$	1	20	4758	4758	4764	75	37	68
$CG + LS$	1	40	4726	4726	4725	326	233	319

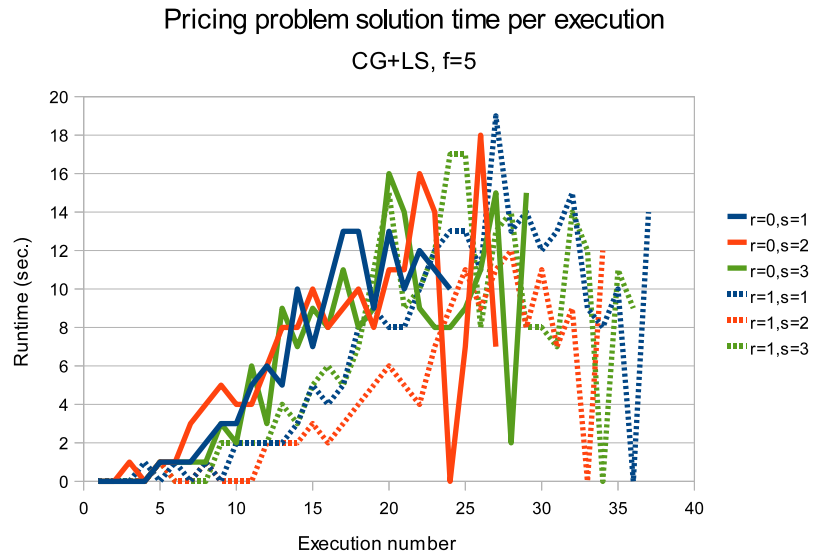
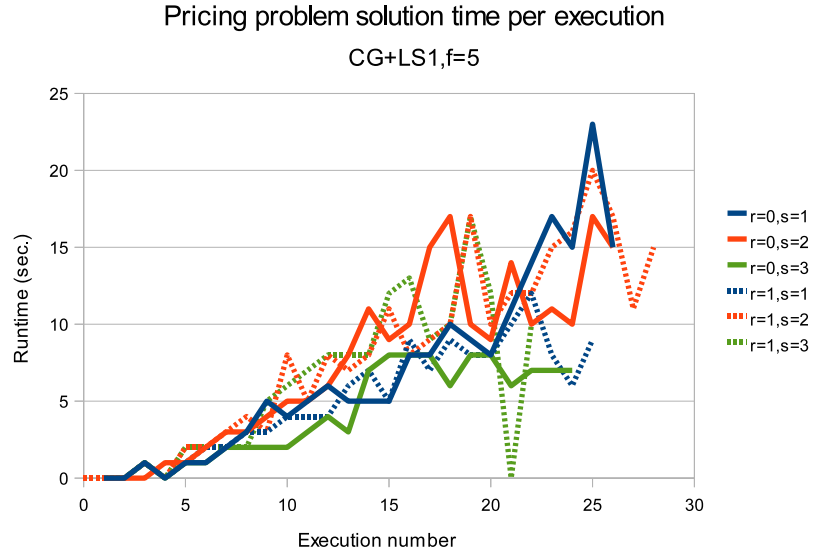
and seed parameters are reported. As expected, pricing problem execution time increases as the number of executions increase, because the number of columns increase. In these experiments  $EN$  was set to 40, but the neighborhood search procedure ceased to find columns to add prior to execution number 40 in each experiment. This explains the slight decrease in pricing problem solution time nearing the final execution of neighborhood search in each experiment.

#### 4.6.4 Results for 10-nurse instance, $TSP$

##### Solution quality

In this set of experiments, 16 districts are designed for teams of  $f = 10$  nurses, and workload bounds are set at [1900,2320] minutes.  $BI$  is able to provide an initial feasible solution for this instance. Table 16 reports the best solutions obtained by  $BI$ ,  $LS$ ,  $CG + BI$ , and  $CG + LS$ . Each value reported corresponds to a single observation. The best solution obtained for this instance has total cost 6764, and is provided by  $CG + BI$ . It is interesting to note that  $CG + BI$  was able to provide a better solution than  $CG + LS$ , despite having been passed an initial solution of lesser quality. Perhaps the dual values associated with solutions of lesser quality provide more guidance to the neighborhood search procedure than those associated with solutions that have already been improved via local search.

The parameters for  $CG + BI$  which were used to obtain the best solution were  $r = 0$ ,  $s = 2$ , and  $EN = 40$ . Table 17 gives the number of moves of each type which were implemented in the corresponding experiment, and the number of districts corresponding



**Figure 21:** Pricing problem execution time,  $TSP$ ,  $f = 5$

**Table 16:** Best solution obtained using each heuristic,  $TSP$ ,  $f = 10$ 

	Sol Val	% imp. over $BI$	% imp. over $LS$
$BI$	7188	-	-
$LS$	7039	2.1%	-
$CG + BI$	6764	5.9%	3.9%
$CG + LS$	6779	5.7%	3.7%

to each type of move which were selected in the final solution to  $P$ . Note that none of the districts in the initial set provided by  $BI$  remain in the final solution. As in the 5-nurse instance, the type of move implemented most frequently is append/reject. However, in this instance, the type of move resulting in the most districts selected in the final solution is swap.

**Table 17:** Moves of each type implemented and selected,  $TSP$ ,  $f = 10$ 

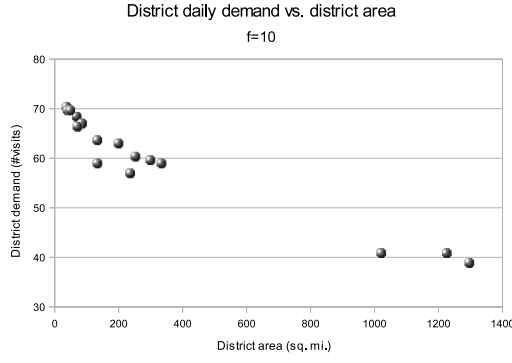
	A	R	A/R	S	I	Total
# implemented	391	2645	8147	3309	-	14508
# selected in final solution	3	0	6	7	0	16

For the best solution obtained for this instance, Figure 22 displays district daily demand, district daily workload, and routing cost as a function of district size. As in the 5-nurse instance, the daily demand included in a single district decreases as district size increases, and the routing cost, or travel time required to serve daily demand, increases. Larger districts again tend to have workloads near the lower bounds.

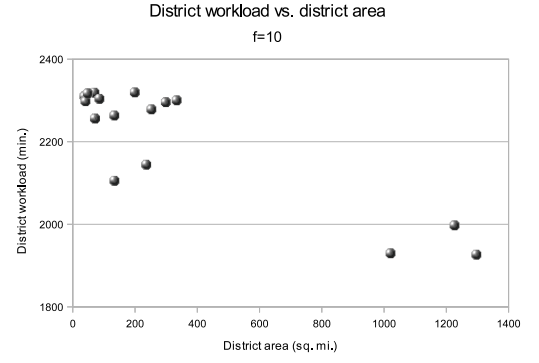
The district maps associated with the best solutions achieved by  $BI$ ,  $LS$ , and  $CG + BI$  are given in Figures 27 - 29 at the end of this chapter. The districts in solutions produced using  $LS$  and  $CG + BI$  have less regular shape than the solutions produced by  $BI$ . This effect seems to be more pronounced in the 10-nurse instance; perhaps because the ratio of zip code workload to district workload bounds is smaller in this instance. This has the effect of permitting a larger number of moves at each execution of neighborhood search.

### Column generation procedure performance

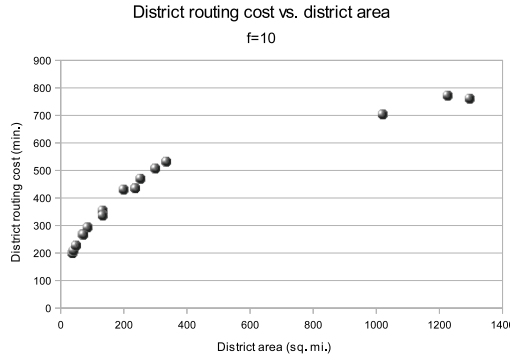
Table 18 displays the solution value observed and the total runtime for each combination of parameters. For any combination of parameters for which the column generation procedure failed to return the best solution in six hours or less, the results are not reported.  $CG + BI$  clearly performs best on this instance. It achieves better solutions than  $CG + LS$



(a) District demand vs. area



(b) District workload vs. area



(c) District cost vs. area

**Figure 22:** District demand, workload, and cost vs. district area,  $TSP$ ,  $f = 10$

when 10, 20, and 40 executions of the pricing problem are allowed before solving the final integer program. When 10 executions are allowed,  $CG + BI$  with warmup produces a solution with value 6931 in 1.5 minutes. This is 3.6% less than the initial feasible solution value, and 1.5% less than the best solution achieved using  $LS$ . When 20 executions of the pricing problem are allowed,  $CG + BI$  with no warmup produces a solution with value 6834 (4.9% lower than  $BI$ ) and requires a total of 7 minutes of computation time. A solution with similar value (6836) is provided by  $CG + LS$  with warmup, but in twice the run time. Achieving the best solution value of 6764, which represents a 5.9% improvement over the initial feasible solution, requires approximately 30 minutes of computation time. Only 15 of those seconds are attributed to IP solve time. However, other combinations of run parameters can require more computation time. For example, when  $CG + BI$  with

warmup is allowed 40 executions of the pricing problem using random seed  $s = 3$ , 26 hours of computation time were required, 25 hours of which were devoted to solving the IP.

**Table 18:** Column generation solution values and runtimes,  $TSP$ ,  $f = 10$

Method	$r$	$EN$	Solution value			Runtime (sec.)		
			$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$CG + BI$	0	10	7015	6946	6970	233	81	98
$CG + BI$	0	20	6864	6834	6903	563	431	588
$CG + BI$	0	40	6835	6764	6848	6409	1560	10775
$CG + BI$	1	10	6931	6999	6962	86	128	103
$CG + BI$	1	20	6863	6890	6925	569	631	556
$CG + BI$	1	40	6810	6840	-	3217	3163	-
$CG + LS$	0	10	6972	6981	7016	98	90	171
$CG + LS$	0	20	6887	6875	6909	528	664	3416
$CG + LS$	0	40	-	6803	6821	-	3321	8988
$CG + LS$	1	10	7028	6952	6987	123	111	98
$CG + LS$	1	20	6978	6879	6836	2623	483	846
$CG + LS$	1	40	6820	6820	6780	8808	4170	2939

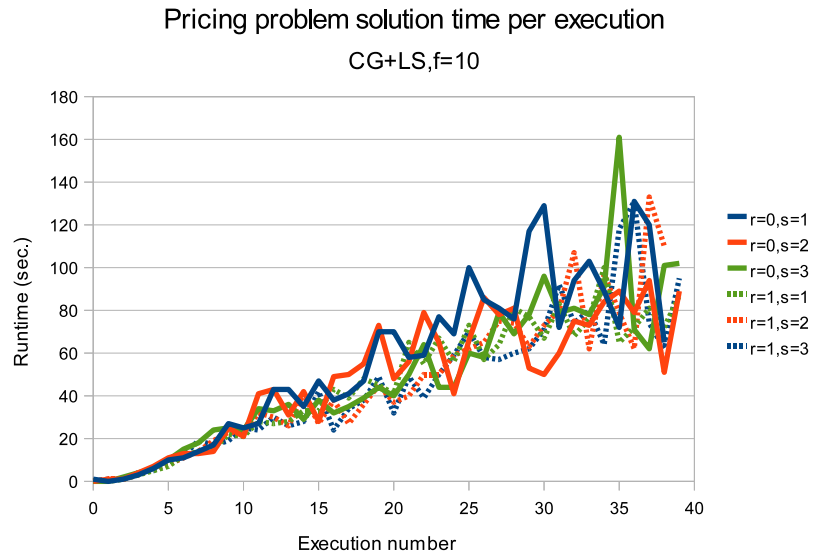
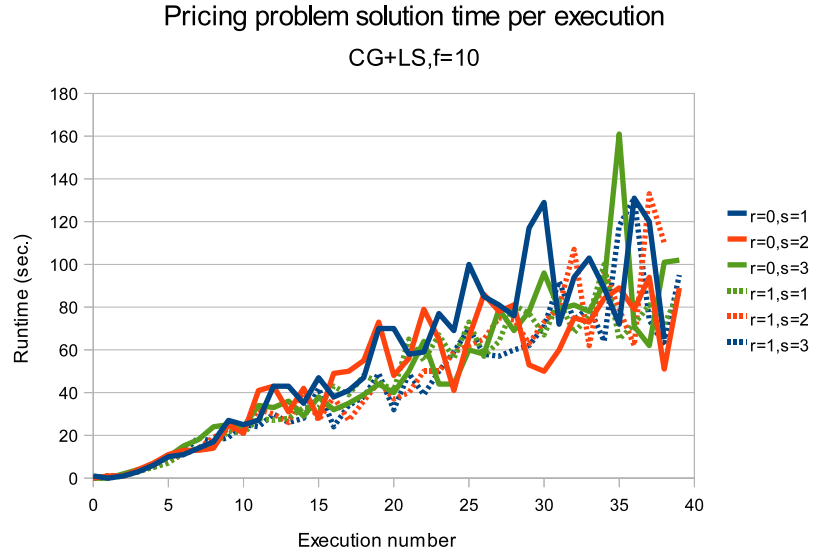
Figure 23 displays the pricing problem execution time in seconds per execution for  $CG + BI$  and  $CG + LS$ , respectively. Observed values for each combination of parameters are reported. Note that more computation time is required in the 10-nurse instance than in the 5-nurse instance. Also, the neighborhood search procedure does not cease to find columns to add before 40 executions of neighborhood search conclude because a larger number of moves are feasible at each execution than in the 5-nurse instance.

#### 4.6.5 Results for 5-nurse instance, $SEP$

In this set of experiments, 32 districts are designed for teams of  $f = 5$  nurses. Workload bounds are set at [955,1200] minutes. Because the  $SEP$  approximations were consistently higher than simulated costs by a factor of 1.75 in initial experiments, travel cost approximations were divided by 1.75 for use in workload estimates. A different set of patient addresses than those used in this instance may require a different scaling factor.

#### Solution quality

Table 19 reports the best solutions obtained by  $BI$ ,  $LS$ ,  $CG + BI$ , and  $CG + LS$ . The initial feasible solution provided by  $BI$  has total cost 5061. The best solution obtained for this instance with total cost 4591 is provided by  $CG + BI$  using parameters  $s = 3$ ,  $r = 0$ , and  $EN = 20$ . Table 20 gives the number of moves of each type implemented and selected in the experiment which provided the best solution for this instance. The type of move both



**Figure 23:** Pricing problem execution time,  $TSP$ ,  $f = 10$

implemented most frequently, and having associated districts selected most frequently in the final solution is append/reject. One third of the initial districts provided by *BI* remain in the final improved solution.

**Table 19:** Best solution obtained using each heuristic, *SEP*,  $f = 5$

	Sol Val	% imp. over <i>BI</i>	% imp. over <i>LS</i>
<i>BI</i>	5061	-	-
<i>LS</i>	4823	4.7%	-
<i>CG + BI</i>	4591	9.3%	4.8%
<i>CG + LS</i>	4823	4.7%	0%

**Table 20:** Moves of each type implemented and selected, *SEP*,  $f = 5$

	A	R	A/R	S	I	Total
# implemented	134	43	853	162	32	1224
# selected in final solution	3	2	14	1	12	32

The district maps associated with the best solutions achieved by *BI*, *LS*, and *CG + BI* are given in Figures 30 - 32 at the end of this chapter. The districts provided when approximation method *SEP* is used tend to be more regular in shape than those provided when approximation method *TSP* is used.

### Routing cost approximation quality

Results from the simulation study summarized in Table 21 reveal that *SEP* does a better job of tracking simulated routing costs than approximation method *TSP*. The lowest cost solution with respect to the *SEP* approximation *and* the simulated costs for this instance is provided by *CG+BI*.

**Table 21:** Comparison of simulated hamiltonian path costs with *SEP* approximation for heuristic solutions,  $f = 5$

Heuristic	Simulated cost	<i>SEP</i> approx.
<i>BI</i>	5101	5061
<i>LS</i>	4871	4823
<i>CG+BI</i>	4803	4591

Recall that the *SEP* travel estimate is scaled down by a factor of 1.75 for use in workload constraints, thus districts feasible with respect to workloads approximated using *SEP* should also be feasible with respect to simulated workloads. This is confirmed in Table

22, which reports both approximated and simulated workloads, and their absolute and percent difference for each district. The mean absolute difference between approximated and simulated workloads is 11.92 minutes, and the mean absolute percent error is 1.1%.

**Table 22:** Comparison of simulated district costs and workloads with *SEP* approximation for *CG+BI* solution,  $f = 5$

$j$	District		<i>SEP</i> approx.		Simulated			Workload diff.	
	Area	Addresses	Cost	Work	Cost ( $\mu$ )	Cost ( $\sigma$ )	Work	Abs.	%
1	478.5	98	327.9	1077.9	352.3	45.9	1102.3	24.4	2.2
2	525.6	43	301.1	1051.1	324.9	34.8	1074.9	23.8	2.2
3	239.3	89	155.3	1055.3	146.8	18.3	1046.8	8.5	-0.8
4	143	61	143.0	1043	156.2	18.4	1056.2	13.2	1.2
5	38.8	39	102.2	1002.2	97.5	11.9	997.5	4.7	-0.5
6	131.5	70	126.3	1026.3	138.6	16.9	1038.6	12.3	1.2
7	86.7	38	165.0	1065.1	173	14.3	1073	7.9	0.7
8	10.6	28	58.7	1108.7	54.3	6.4	1104.3	4.4	-0.4
9	15.5	60	60.7	960.7	63.6	6.9	963.6	2.9	0.3
10	93.5	20	147.0	1047	133.8	15.2	1033.8	13.2	-1.3
11	46.6	43	127.1	1027.1	126.1	15.7	1026.1	1	-0.1
12	22.4	62	73.0	973	77.7	7.6	977.7	4.7	0.5
13	83	88	127.7	1177.6	129	14	1179	1.4	0.1
14	21.7	19	87.0	1136.9	68.5	6.9	1118.5	18.4	-1.6
15	19.2	40	81.7	981.7	80.1	10.8	980.1	1.6	-0.2
16	392.9	80	228.3	978.3	258.9	29.9	1008.9	30.6	3
17	217.7	52	215.1	1115.1	240	24.2	1140	24.9	2.2
18	29.1	54	81.8	981.8	84.8	7.5	984.8	3	0.3
19	23.6	60	84.8	984.8	88.3	10.7	988.3	3.5	0.4
20	620.7	71	218.2	968.2	242.1	27.6	992.1	23.9	2.4
21	832.9	44	209.4	1109.4	214.9	53.4	1114.9	5.5	0.5
22	37.1	67	110.3	1010.3	115.1	12.4	1015.1	4.8	0.5
23	33.5	52	89.3	989.3	92	13.6	992	2.7	0.3
24	70.7	104	109.4	1009.4	118.1	13.2	1018.1	8.7	0.9
25	34.1	17	86.6	986.7	63.3	11.8	963.3	23.4	-2.4
26	642.8	130	257.3	1007.3	293.6	41.8	1043.6	36.3	3.5
27	32.1	31	74.5	974.4	64.5	8.7	964.5	9.9	-1
28	211.6	107	209.6	1109.6	223	27	1123	13.4	1.2
29	123	78	166.0	1066	195	16.2	1095	29	2.6
30	71.9	53	118.5	1018.5	127	17.1	1027	8.5	0.8
31	35.5	25	87.1	987.1	87.3	9.8	987.3	0.2	0
32	118.7	62	162.1	1062.1	172.9	20.3	1072.9	10.8	1
			4591.9		4803.4				

### Column generation procedure performance

Table 23 displays the solution value observed and the total runtime in seconds of the column generation procedure for each combination of parameters. Results are not reported for *CG + LS* because the solutions returned do not improve the *LS* solutions, but recall that the solution provided by *CG+BI* has lower cost than the solution provided by *LS*. Runtimes in this set of experiments remain under six minutes.

To test the impact of the parameter  $\alpha$ , a small set of experiments were performed with  $\alpha = 0.15$  and workload bounds [920,1250]. For this instance, *LS* provided a solution with



**Table 23:** Column generation solution values and runtimes,  $SEP$ ,  $f = 5$ 

Method	$r$	$EN$	Solution value			Runtime (sec.)		
			$s = 1$	$s = 2$	$s = 3$	$s = 1$	$s = 2$	$s = 3$
$CG + BI$	0	20	4862	4862	4858	57	57	66
$CG + BI$	0	40	4862	4862	4591	119	120	165
$CG + BI$	0	60	4862	4862	4591	182	181	253
$CG + BI$	0	80	4862	4862	4591	245	310	350
$CG + BI$	1	20	4830	4832	4795	56	60	64
$CG + BI$	1	40	4830	4666	4795	110	131	142
$CG + BI$	1	60	4830	4666	4795	161	194	219
$CG + BI$	1	80	4830	4666	4795	278	343	295

cost 4797, and  $CG+LS$  provided an improved solution with cost 4637. The best solution was provided by  $CG+BI$  with cost 4530. Thus, setting  $\alpha$  to 0.15 enables a solution with 1.3% lower cost than when  $\alpha = 0.10$ .

#### 4.6.6 Absolute performance of the heuristics

The discussion of results has focused on the relative performance of the heuristics. Regarding their absolute performance, we do not have an exact bound on the optimal value of the solutions. We are able to conclude that the best solutions found for the 5-nurse instance are locally optimal. For the 10-nurse instance,  $LS$  was able to improve the best solution provided by  $CG + LS$  by an additional 0.1%. The simulation study validated routing cost approximation method  $SEP$  but not method  $TSP$ .

### 4.7 Conclusions

Routing cost approximation method  $SEP$  was shown to do a better job of tracking simulated travel costs than approximation method  $TSP$ . The poor performance of  $TSP$  is likely attributed to the small number of points visited by each nurse each day. The estimate in Equation (74) is an asymptotic result, but only five to seven patient addresses are included in each path. Thus, approximation method  $TSP$  should not be used in applications like these unless the number of points visited per nurse per day is significantly higher.

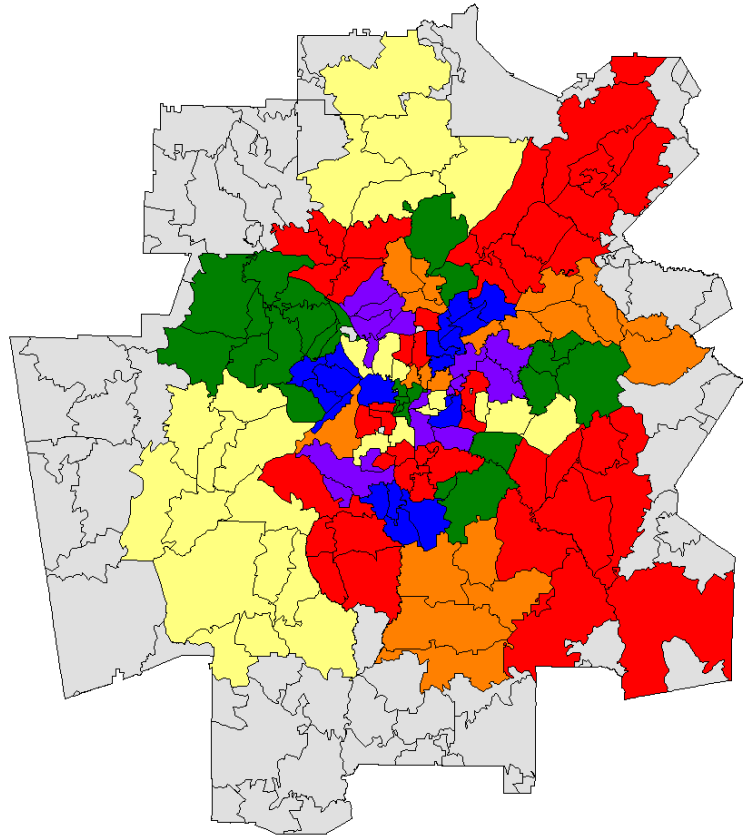
The column generation heuristic developed in this chapter combines ideas from optimization based approaches and heuristic local search approaches to solve a set partitioning model for the home health nurse districting problem. This approach demonstrates that

some of the benefit typically associated with optimization approaches such as branch-and-price can be realized in the context of a heuristic approach. The column generation heuristic is shown to be effective in improving both initial feasible solutions provided by a heuristic construction method and those provided by a local search improvement method.

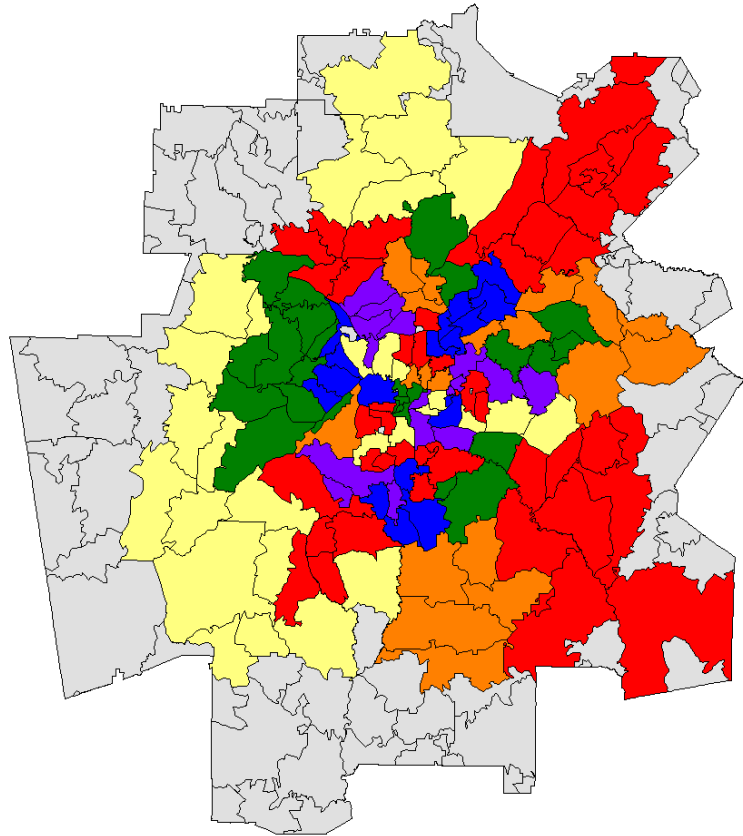
Improvements of 5% over the initial feasible solution provided to the column generation heuristic are obtained using 10 minutes or less computation time in most instances. When additional computation time is available, allowing a larger number of executions of neighborhood search may enable additional improvements; however, it is not clear what combination of parameters results in a final integer program that solves in a reasonable amount of time in all instances. If a better solution than what can be obtained using a small number of neighborhood search executions is desired, a reasonable strategy is to include a parameter which indicates the method should return the best solution identified after a pre-specified amount of time.

The primary benefit of the method developed in this chapter is the ability to quickly obtain good solutions to various scenarios. By setting the neighborhood search execution parameter  $EN$  to a small number such as 10 to 20 executions, a user can experiment with various nurse team sizes and workload balance parameters to determine preferred scenarios. Then if desired, the execution limit can be increased to allow the method more time to obtain an improved solution for the selected scenario.

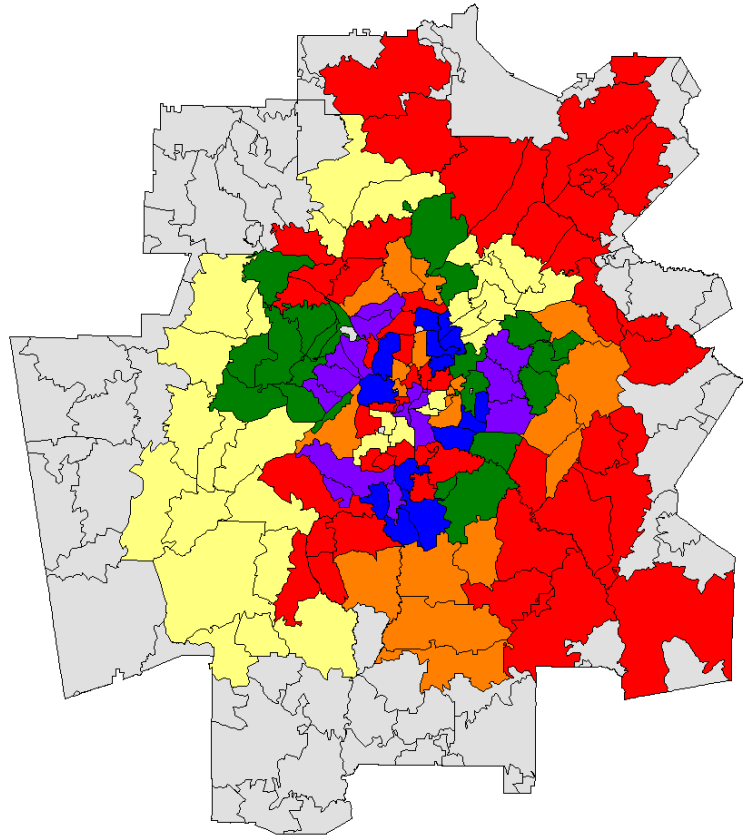
The method can also be used to provide insight to various managerial decisions, such as determining portions of the service region in which it may be beneficial to focus growth efforts. Home health agencies often develop partnerships with hospitals to increase the number of home health care referrals they receive for patients being discharged from the hospital. The impact of increasing patient enrollment in various geographic regions can be assessed by using the method with simulated data sets that reflect increased demand levels in specified regions. Additionally, the method can be easily modified so that the impact of balancing patient visit count or patient caseloads, which incorporate variations in patient service times, can be determined.



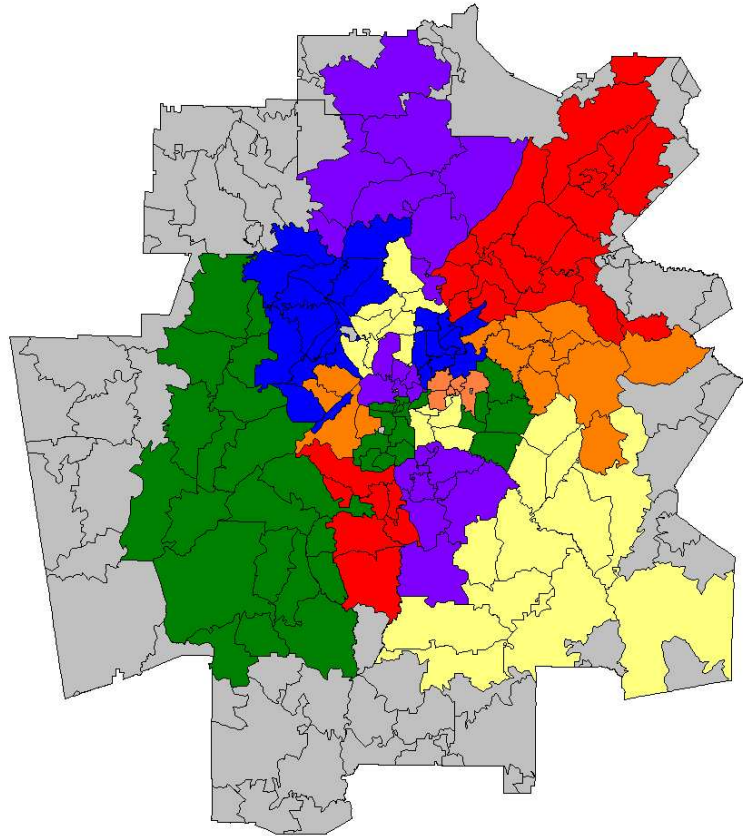
**Figure 24:** *LS1* solution map, *TSP*,  $f = 5$



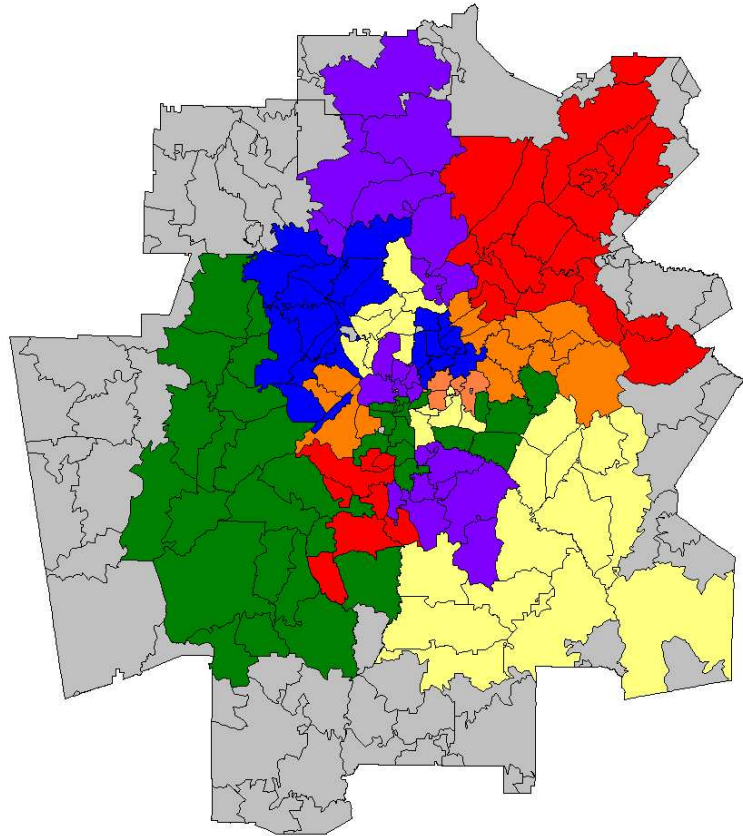
**Figure 25:** *LS* solution map, *TSP*,  $f = 5$



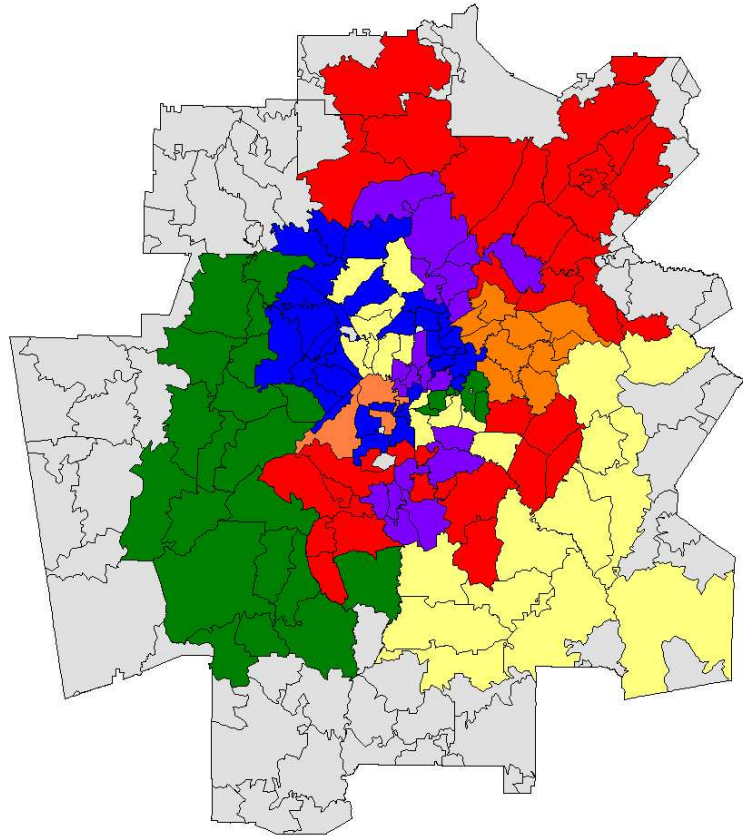
**Figure 26:**  $CG + LS$  solution map,  $TSP$ ,  $f = 5$



**Figure 27:** *BI* solution map, *TSP*,  $f = 10$

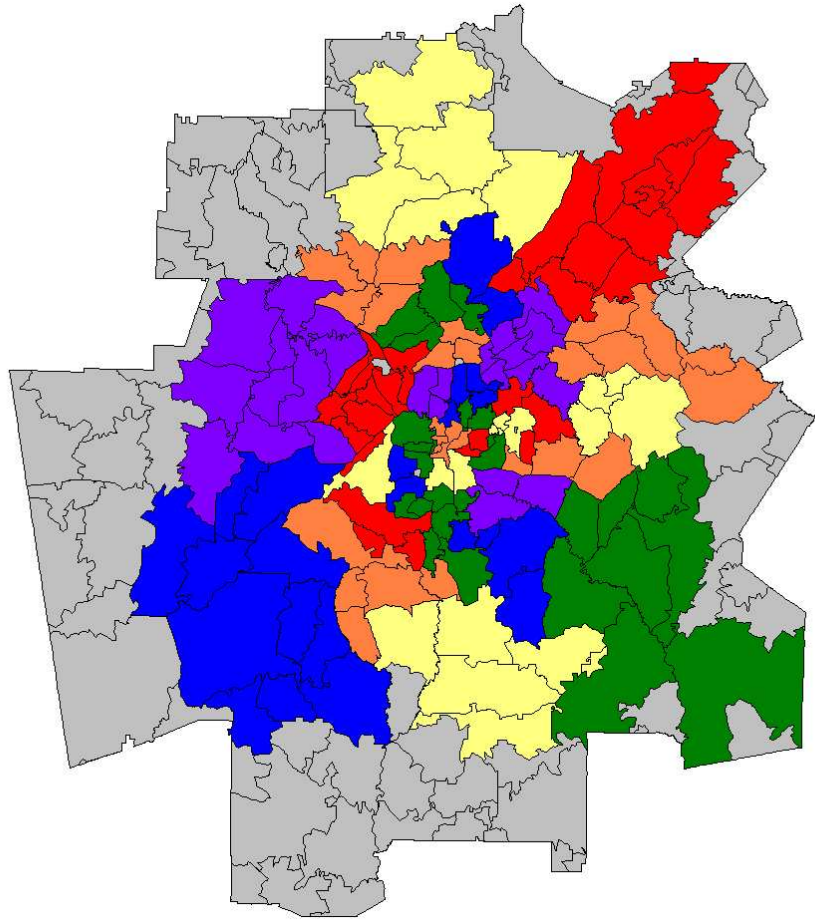


**Figure 28:** *LS* solution map, *TSP*,  $f = 10$

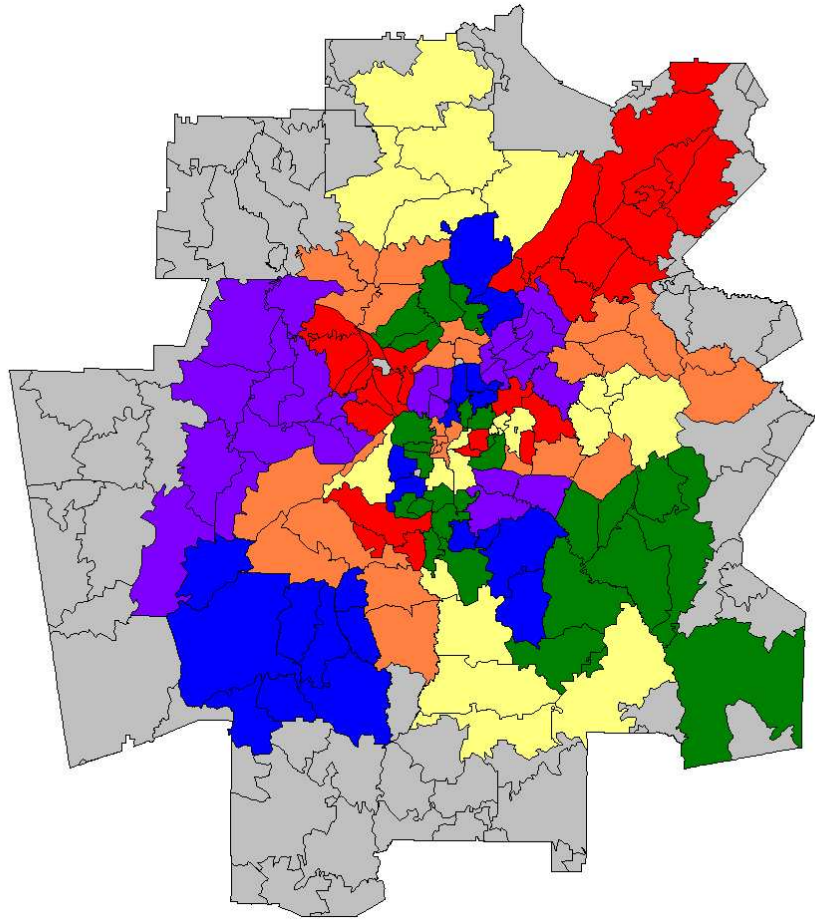


**Figure 29:** *CG + BI* solution map, *TSP*,  $f = 10$

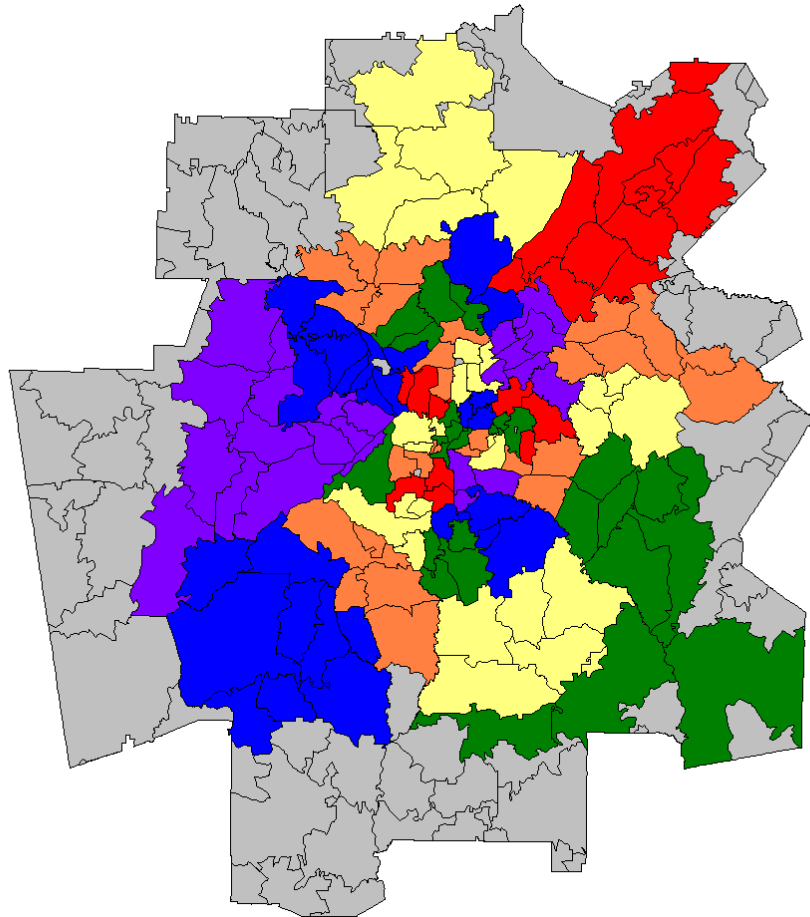




**Figure 30:** *BI* solution map, *SEP*,  $f = 5$



**Figure 31:** *LS* solution map, *SEP*,  $f = 5$



**Figure 32:**  $CG + BI$  solution map,  $SEP$ ,  $f = 5$

## CHAPTER V

### FUTURE RESEARCH DIRECTIONS

This thesis provides solution methods for both tactical and operational home health logistics planning problems. In Chapter 3, a rolling horizon myopic planning approach is developed to solve the operational problem of incorporating appointments for newly arrived patient requests into daily visit schedules for a single nurse. In Chapter 4, an optimization-based local search heuristic is developed to solve the tactical problem of assigning home health nurses to geographic service regions. Avenues for future research specific to each approach were identified in the respective chapters. In this chapter, possibilities for integrating the two approaches are discussed.

Suppose each district in the solution to the home health nurse districting problem is served by a single nurse. Then, the rolling horizon myopic planning approach can be used to develop visit schedules for the nurse serving each district. However, if fixed appointment time constraints are enforced and no flexibility in visit times is allowed from week to week, 10 to 20 percent of the patients requesting service in each district may be denied service. In practice, home health agencies would not turn away this volume of business. The alternatives to denying service to patients due to the rigid problem structure are to relax some subset of the fixed appointment, visit repeatability, or provider continuity constraints.

The fixed appointment constraints could be relaxed by specifying a time window during which the patient can expect a visit instead of specifying an exact time. Visit repeatability constraints could be enforced in this setting by requiring the time window for each visit to be the same from week to week. As an alternative to relaxing fixed appointment constraints, visit repeatability constraints could be relaxed, allowing the appointment time to vary within some interval from week to week. For example, a Wednesday appointment could be allowed to occur anytime between 8:00 a.m. and 10:00 a.m. each week, as long as an exact appointment time were specified for each visit. Developing nurse schedules for either case

requires extensions to the scheduling method in developed in Chapter 3.

Extending the method developed in Chapter 3 to the multiple-nurse scenario would allow developing schedules for which provider continuity constraints are relaxed. The extended method could then be used in combination with districts specified by the home health nurse districting method for instances with  $f > 1$ . Having the ability to pass solutions between HHNRS and HHND solution methods may enable a routing cost estimate appropriate for the dynamic periodic setting to be developed. Ultimately, with full integration of methods, it would be possible to determine which of the fixed appointment, visit repeatability, and provider continuity constraints have the most impact on nurse productivity. Information regarding which of the constraints are most preferred by home health patients would be useful in guiding future home health logistics planning research efforts.

## Bibliography

- [1] National and regional estimates on hospital use for all patients from the HCUP nationwide inpatient sample (NIS): 2007 outcomes by patient and hospital characteristics for all discharges. Available from the Agency for Healthcare Research and Quality (AHRQ) at <http://hcupnet.ahrq.gov/HCUPnet.jsp>. Accessed November 4 2009.
- [2] UPS corporate responsibility: a commitment to safety, every day. Available at <http://responsibility.ups.com/safety/index.html>. Accessed November 3 2009.
- [3] C. Akjiratikarl, P. Yenradee, and P.R. Drake. PSO-based algorithm for home care worker scheduling in the UK. *Computers and Industrial Engineering*, 53:559–583, 2007.
- [4] S. Baptista, R.C. Oliveira, and E. Zuquete. A period vehicle routing case study. *European J. Operational Research*, 139:220–229, 2002.
- [5] J. Beardwood, J. Halton, and J. Hammersley. The shortest path through many points. *Proceedings of the Cambridge Philosophical Society*, 55:299–327, 1959.
- [6] S.V. Begur, D.M. Miller, and J.R. Weaver. An integrated spatial DSS for scheduling and routing home health care nurses. *Interfaces*, 27:35–48, 1997.
- [7] R.W. Bent and P. van Hentenryck. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.
- [8] L. Bertazzi, G. Paletta, and M.G. Speranza. An improved heuristic for the period traveling salesman problem. *Computers and Operations Research*, 31:1215–1222, 2004.
- [9] S. Bertels and T. Fahle. A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Computers and Operations Research*, 33:2866–2890, 2006.
- [10] D.E. Blumenfeld and M.J. Beckmann. Use of continuous space modeling to estimate freight distribution costs. *Transportation Research A*, 19(2):173–187, 1985.
- [11] B. Bozkaya, E. Erkut, and G. Laporte. A tabu search heuristic and adaptive memory procedure for political districting. *European J. Operational Research*, 144:12–26, 2003.
- [12] P.I. Buerhaus, D.O. Staiger, and D.I. Auerbach. Implications of an aging registered nurse workforce. *Journal of the American Medical Association*, 283:2948–2954, 2000.
- [13] Michael D. Cabana and Sandra H. Jee. Does continuity of care improve patient outcomes. *The Journal of Family Practice*, 53(12), 2004.
- [14] F. Caro, T. Shirabe, M. Guignard, and A. Weintraub. School redistricting: embedding gis tools with integer programming. *Journal of the Operational Research Society*, 55: 836–849, 2004.
- [15] I.M. Chao, B.L. Golden, and E.A. Wasil. A new heuristic for the period traveling salesman problem. *Computers and Operations Research*, 22(5):553–565, 1995.

- [16] N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
- [17] CMS. Home health quality initiatives. Technical report, Centers for Medicare and Medicaid Services, 7500 Security Boulevard, Baltimore MD, 21244, 2008. Available at <http://www.cms.hhs.gov/homehealthQualityInits/>.
- [18] William J. Cook. Concorde tsp solver. Available at <http://www.tsp.gatech.edu>.
- [19] J-F. Cordeau, M. Gendreau, and G. Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, 30:105–119, 1997.
- [20] S. D’Amico, S-J. Wang, R. Batta, and C. Rump. A simulated annealing approach to police district design. *Computers and Operations Research*, 29:667–684, 2002.
- [21] P. Eveborn, P. Flisberg, and M. Ronnqvist. LAPS CARE - an operational system for staff planning of home care. *European J. Operational Research*, 171:962–976, 2006.
- [22] D. Feillet, P. Dejax, and M. Gendreau. Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205, 2005.
- [23] P. Francis, K. Smilowitz, and M. Tzur. The period vehicle routing problem with service choice. *Transportation Science*, 40(4):439–454, 2006.
- [24] R.S. Garfinkel and G.L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16(8):B495–B508, 1970.
- [25] D. Haugland, S. Ho, and G. Laporte. Designing delivery districts for the vehicle routing problem with stochastic demands. *European J. Operational Research*, 180:997–1010, 2007.
- [26] V.C. Hemmelmayr, K.F. Doerner, and R.F. Hartl. A variable neighborhood search heuristic for periodic routing problems. *European J. Operational Research*, 195(3): 791–802, 2009.
- [27] S. W. Hess and S. A. Samuels. Experiences with a sales districting model: criteria and implementation. *Management Science*, 18(4):41–54, 1971.
- [28] S.W. Hess, J.B. Weaver, H.J. Siegfeldt, J.N. Whelan, and P. A. Zitlau. Non-partisan political redistricting by computer. *Operations Research*, 13:998–1006, 1965.
- [29] M. Hojati. Optimal political districting. *Computers and Operations Research*, 23(12): 1147–1161, 1996.
- [30] Susan L. Hughes, Alec Ulasevich, Frances M. Weaver, William Henderson, Larry Mannheim, Joseph D. Kubal, and Frank Bonarigo. Impact of home care on hospital days: a meta analysis. *Health Services Research*, 32(4):415–432, 1997.
- [31] L.M. Hvattum, A. Lokketangen, and G. Laporte. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. *Transportation Science*, 40(4):421–438, 2006.

- [32] S. Ichoua, M. Gendreau, and J.Y. Potvin. Exploiting knowledge about future demands for real-time vehicle dispatching. *Transportation Science*, 40(2):211–225, 2006.
- [33] D.S. Johnson, L.A. McGeoch, and E.E. Rothberg. Asymptotic experimental analysis for the held-karp traveling salesman bound. *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 341–350, 1996.
- [34] G. Laporte and S. Martello. The selective traveling salesman problem. *Discrete Applied Mathematics*, 26:193–207, 1990.
- [35] A. Mehrotra, E. Johnson, and G. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114, 1998.
- [36] NAHC. Basic statistics about home care. Technical report, National Association for Home Care and Hospice, 228 Seventh St SE, Washington DC, 20033, 2007. Available at <http://www.nahc.org/>.
- [37] NAHC. Study shows home health care workers drive nearly five billion miles to serve elderly and disabled patients. Technical report, National Association for Home Care and Hospice, 228 Seventh St SE, Washington DC, 20033, 2009. Available at <http://www.nahc.org/facts/homecareStudy.html>.
- [38] M. Polacek, K.F. Doerner, R.F. Hartl, G. Kiechle, and M. Reimann. Scheduling periodic visits for a traveling salesperson. *European J. Operational Research*, 179:823–837, 2007.
- [39] F. Ricca and B. Simeone. Local search algorithms for political districting. *European J. Operational Research*, 189:1409–1426, 2008.
- [40] Jennifer Rich. A computational study of vehicle routing applications. PhD dissertation, Rice University, 1999.
- [41] Takeshi Shirabe. A model of contiguity for spatial unit allocation. *Geographical Analysis*, 37:2–16, 2004.
- [42] Jorg Michael Steeg. Mathematical models and algorithms for home care services. PhD dissertation, Fraunhofer Institute for Industrial Mathematics, 2008.
- [43] Nora Super. Who will be there to give care? The growing gap between caregiver supply and demand. White paper, National Health Policy Forum of The George Washington University, Washington DC, January 2002.
- [44] J.I. van Hemert and J.A. LaPoutre. Dynamic routing problems with fruitful regions: models and evolutionary computation. *Parallel Problem Solvers from Nature*, 8:690–699, 2004.
- [45] A. Zoltners and P. Sinha. Sales territory alignment: a review and model. *Management Science*, 29(11):1237, 1983.